TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií

MODUL PRO PODPORU PROCESORŮ Z ŘADY AVR NA EXISTUJÍCÍM VÝUKOVÉM PŘÍPRAVKU

Bakalářská práce

M11000162

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Autor práce: **Josef Kracík**
Vedoucí práce: Ing. Tomáš Martinec, Ph.D.

Liberec 2014

TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies

MODULE WITH AVR PROCESSOR FOR EXISTING EDUCATION KIT

Bachelor thesis

M11000162

Study programme: B2612 – Electrical Engineering and Informatics
Study branch: 2612R011 – Electronic Information and Control Systems

Author: **Josef Kracík**
Supervisor: Ing. Tomáš Martinec, Ph.D.

Liberec 2014

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky a mezioborových studií
Akademický rok: **2013/2014**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Josef Kracík**
Osobní číslo:	**M11000162**
Studijní program:	**B2612 Elektrotechnika a informatika**
Studijní obor:	**Elektronické informační a řídicí systémy**
Název tématu:	**Modul pro podporu procesorů z řady AVR na existujícím výukovém přípravku**
Zadávající katedra:	**Ústav mechatroniky a technické informatiky**

Zásady pro vypracování:

1. Seznamte se s řadou procesorů AVR a možnostmi jejich využití s existujícím výukovým přípravkem.
2. Navrhněte modul s vybraným procesorem z řady AVR pro použití s výukovým přípravkem.
3. Realizujte tento modul a otestujte jeho funkci na vhodných příkladech.

Rozsah grafických prací: **dle potřeby dokumentace**

Rozsah pracovní zprávy: **30–40 stran**

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

[1] **Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, Morgan Skinner: C# 2008, Programujeme profesionálně, Computer Press, duben 2009, ISBN: 978-80-251-2401-7**

[2] **David Matoušek: Práce s mikrokontroléry Atmel AVR, BEN - technická literatura, červen 2006, ISBN 80-7300-209-4**

Vedoucí bakalářské práce: **Ing. Tomáš Martinec, Ph.D.**
Ústav mechatroniky a technické informatiky

Konzultant bakalářské práce: **Ing. Josef Grosman**
Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: **10. října 2013**

Termín odevzdání bakalářské práce: **16. května 2014**

prof. Ing. Václav Kopecký, CSc.
děkan

L.S.

doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2013

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Datum:

Podpis:

Poděkování

Rád bych poděkoval Ing. Tomáši Martincovi, Ph.D. za ochotu, odborné rady a cenné připomínky, které výrazně přispěly k vypracování této bakalářské práce.

Abstrakt

Předmět Počítače a mikropočítače z Ústavu mechatroniky a technické informatiky využívá výukový přípravek, který seznamuje studenty se základními principy technického vybavení počítačů. Tento přípravek je vybaven různými komunikačními obvody, displejem, klávesnicí, čidly, tlačítky, LED diodami atp. Obsahuje také patici, do které se dá připojit modul s různými typy mikrokontroléru. Tato bakalářská práce popisuje výrobu takového modulu, který bude obsahovat řídící mikrokontrolér specifické řady AVR a dokáže ovládat veškeré periférie přípravku. Svoji použitelnost by měl ukázat na vhodně naprogramovaných příkladech. Výsledný AVR modul by měl být schopen plné funkce při použití ve výuce. Uplatnění tohoto modulu by měli ocenit hlavně začínající programátoři (studenti ostatních fakult), pro které bylo vymyšleno použití speciální platformy, jenž umožňuje snadnější programování.

Klíčová slova:

výukový přípravek, AVR, modul, programování, Arduino

Abstract

Institute of Mechatronics and Computer Engineering includes subject called Computers and Microcomputers which introduces students to the basic principles of computer hardware using an educational kit. This product is a printed circuit board and it is equipped with various communication circuits, display, keypad, sensors, buttons, LEDs, etc. It also includes a special connector for univesal modules which contains different microcontrollers. This bachelor thesis describes main steps of making a special module which will include a specific AVR microcontroller. This MCU will be possible to control the whole educational kit. All its practicability should be shown on an appropriately programmed examples. The final AVR module should be able to work in the subject. The application of this module should be helpful for beginning programmers (especially from another faculties than students of the Faculty of Mechatronics). This AVR module contians a special platform which allows easier programming.

Key words:

educational kit, AVR, module, programming, Arduino

Obsah

Seznam obrázků

Seznam tabulek

Seznam zkratek

ACK	ACKnowledgement signal, potvrzující bit I^2C sběrnice
AVR	Alf (Egil Bogen) Vegard (Wollan) Risc procesor, RISC mikropočítač od firmy Atmel
CAN	sběrnice používaná v automobilovém průmyslu
CPU	Central Procesing Unit, součástka vykonávající strojový kód z operační paměti
CTC	Clear Timer on Compare, speciální mód časovače
DB9	konektor pro sériovou komunikaci RS-232
EEPROM	Electrically Erasable Programmable Read-Only Memory, elektricky mazatelná nonvolatilní paměť
ERC	Electrical Rule Check, kontrola elektrických pravidel ve schématu návrhu plošného spoje
GND	GrouND, označení zemnícího vodiče
GNU	GNU's Not Unix, projekt zaměřený na svobodný software
GPL	General Public License, všeobecná veřejná licence GNU
GUI	Graphical User Interface, grafické uživatelské rozhraní
I^2C	Inter-Integrated Circuit, sériová sběrnice
ISP	In-System Programming, programovací rozhraní
JTAG	Joint Test Action Group, metoda testování plošných spojů a možnost programování Flash pamětí
LCD	Liquid Crystal Display, displej složen z tekutých krystalů
LED	Light-Emitting Diode, polovodičová elektronická součástka vyzařující světlo
MCU	Micro Controler Unit, jiný význam pro mikrokontrolér
MISO	Master In Slave Out, vodič ISP komunikace
MOSI	Master Out Slave In, vodič ISP komunikace
NAND	Not AND, logický člen negovaného součinu
NOR	Not OR, logický člen negovaného součtu
NOT	Not, logický člen negace
PCB	Printed Circuit Board, jiné označení pro desku plošného spoje
PWM	Pulse Width Modulation, pulzně šířková modulace
RGB	Red Green Blue, tři základní barvy červená, zelená a modrá
ROM	Read-Only Memory, paměť určená pouze ke čtení
RS-232	rozhraní sériové komunikace
RS-485	rozhraní sériové komunikace
RTC	Real Time Clock, integrovaný obvod kalkulující reální čas
RWM	Read-Write Memory, paměť umožnující čtení i zápis
SCL	Serial Clock Line, hodinový signál I^2C sběrnice
SCK	Serial ClocK, Master Out Slave In, hodinový signál ISP komunikace
SDA	Serial Data Line, datový signál I^2C sběrnice
SMD	Surface Mount Device, součástky určené k povrchové montáži na plošném spoji
TWI	Two Wire serial Interface, dvouvodičové rozhraní
USB	Universal Serial Bus, univerzální sériová sběrnice
VCC	označení napájecího napětí

1. Úvod

Již v dávných dobách byla snaha lidí ulehčovat si práci různými chytrými vynálezy, které by zmírnily námahu samotného člověka. Jednalo se především o stoje nebo nástroje ovládané manuálně člověkem. Později se začalo uvažovat také o strojích, které by dokázaly samostatně pracovat bez lidské pomoci. Měly by tzv. umělou inteligenci. Centrum řízení stroje v dnešní době tvoří tzv. mikrokontrolér, který ovládá pomocí elektrických signálů jeho jednotlivé části. První mikrokontrolér vytvořila firma Texas Instruments v roce 1974. Tento prvek dnes nalezneme téměř v každém moderním elektronickém přístroji např. v hračkách, mp3 přehrávačích, dálkových ovládačích, mobilních telefonech, lékařských nebo měřících přístrojích, v implantátech a v celé řadě dalších zařízeních.

Jednočipový počítač (MCU, uC, mikrokontrolér, angl. microcontroller) je programovatelná elektronická součástka v podobě integrovaného obvodu. Používá se převážně k řízení nebo jako součást nějakého dalšího zařízení k plnění určité funkce. Obsahuje paměť, programovatelné vstupně-výstupní rozhraní a různé další periferní obvody např. čítače, časovače, sériové porty, analogově-digitální nebo digitálně-analogové převodníky a další. Použitím mikrokontroléru v dnešní době nahradíme velké množství logických obvodů i diskrétních součástek, kterých bylo dříve k realizaci potřeba. Tím pádem se snížily náklady, ale také se rozhraní stalo daleko přívětivějším pro uživatele. Přidáním nového komponentu nebo zařízení tak nemusíme měnit konstrukci samotného jednočipového počítače. Stačí jen pozměnit jeho instrukce tzn. jej přeprogramovat.

Cílem této bakalářské práce je vybrat takový mikrokontrolér od firmy Atmel z řady AVR, který by dokázal ovládat všechny komponenty školního přípravku, a seznámit se s ním z hlediska programování. Obeznámit se také s výukovým přípravkem, především s jeho periférierni a možnostmi použití. Navrhnout plošný spoj nebo-li modul, který bude obsahovat jednak daný mikročip, ale také programovací obvod tzv. programátor, aby se při každém programovaní nemusel připojovat speciální programátor externě. Po výrobě a osazení modulu nahrát do předem připraveného mikročipu (jiného než řídícího) kód, který zaručí stejnou funkci, jakou vykonává externí programátor. Posledním bodem této práce je naprogramování jednotlivých periferií přípravku v jazyce C, a tím otestovat vyrobený modul. Následně zkusit použít Arduino platformu a nahrát s pomocí speciálních knihoven výrazně jednodušší program. Při výuce by tedy mohl začátečník používat právě toto prostředí, a poté postupně přecházet do samostatného programovacího jazyku C.

2. Výukový přípravek

Výukový přípravek je deska plošného spoje (obr. 2.1, str. 15), která obsahuje různé periferní obvody jako např. (LCD displej, klávesnici apod.). Využívá se zejména v předmětu PMP (Počítače a mikropočítače) a PHS (Počítačový hardware a rozhraní).

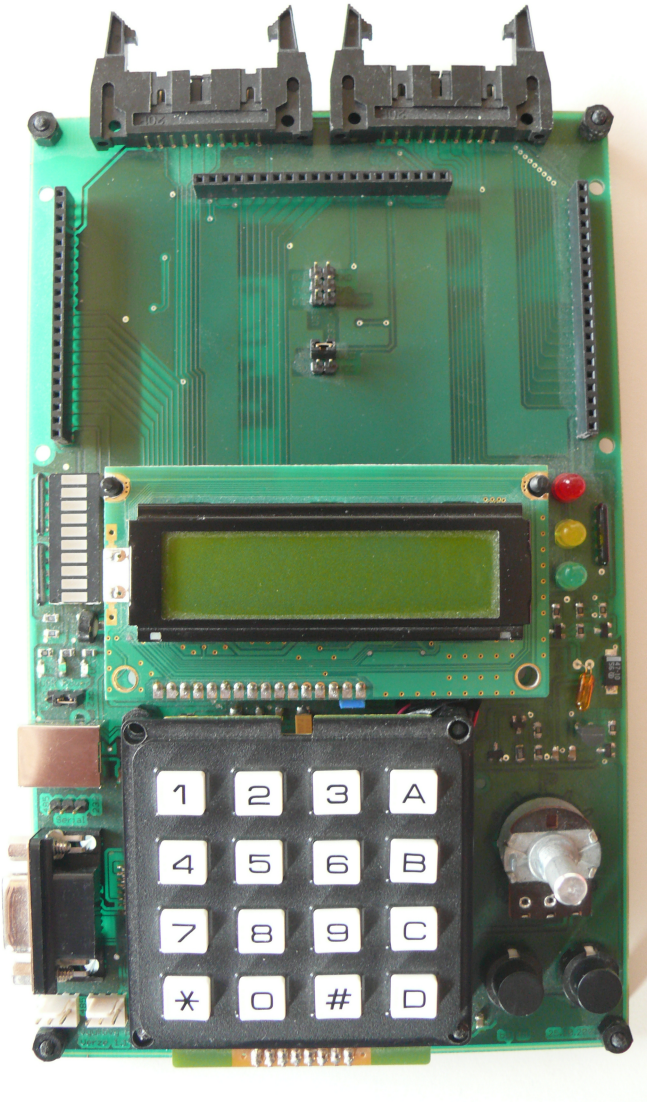
Seznam důležitých komponent přípravku

- **Konektor 2 x 20 pinů** - speciální konektor ML 20, který propojuje periferie přípravku s řídícím modulem, použití závisí na uživateli.
- **Konektor 3 x 20 pinů** - jedná se o tři dutinkové lišty o dvaceti pinech, které mají přesné rozestavení a slouží pro připojení příslušného modulu k výukovému přípravku.
- **LCD displej s řadičem** - displej obsahuje integrovaný obvod HD44780 s rozlišením 16x2 znaků a LED pole jako podsvícení.
- **Maticová klávesnice** - klávesnice, která obsahuje čtyři řádky a sloupce, dohromady tedy šestnáct kláves.
- **LED bargraf** - sloupec 10-ti zelených LED diod v jednom uceleném pouzdru s 20-ti vývody.
- **Senzor teploty** - integrovaný obvod, který představuje moderní křemíkové teplotní čidlo určené pro nejrůznější konstrukce, jehož výstupem je šířkově modulovaný signál.
- **Tlačítko 2x** - tlačítkový spínač bez aretace.
- **Potenciometr** - uhlíkový otočný mono potenciometr.
- **Piezosirénka** - komponenta ovládaná pomocí šířkově modulovaného signálu.
- **LED diody 3x** - trojice LED diod červené, žluté a zelené barvy.
- **Žárovka**

- **USB konektor** - USB typu B (zásuvka), která slouží jako hlavní přívod napájení nebo komunikace po sériové lince pomocí integrovaného obvodu FT232 umístěného na přípravku.

- **DB9** - konektor (zásuvka), využívající se k sériové komunikaci po RS-232 a RS-485 pomocí integrovaných obvodů MAX232 a MAX481, které jsou umístěny na přípravku.

- **Konektor se zámkem 2x** - konektory pro připojení zařízení, které bude komunikovat pomocí RS-485.



Obrázek 2.1: Výukový přípravek

3. Mikrokontrolér

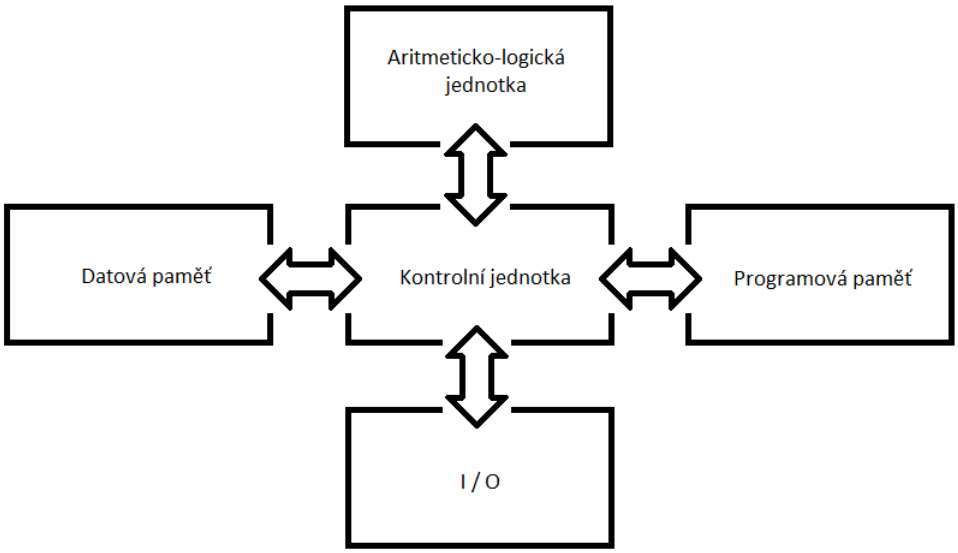
3.1 AVR procesory

AVR je označení pro jednočipový mikrokontrolér s Harvardskou architekturou typu RISC. U zrodu stáli studenti Alf-Egil Bogen a Vegard Wollan z Norského technologického institutu (Norwegian Institute of Technology(NTH)). Název AVR vyplývá z Alf-Egil Bogen Vegard Wollan RISC processor. Firma Amtel koupila tuto koncepci, a tím vznikla 8-bitová MCU rodina, která je na trhu od roku 1997. Byla to také jedna z prvních řad, která měla uložený program v paměti Flash.

3.1.1 Harvardská architektura

Tato architektura se vyznačuje rozdělenou pamětí pro data a pro program. Mezi její hlavní výhody patří možnost odlišné šířky programové a datové sběrnice. Pro program se používá obvykle paměť ROM, protože není tak finančně nákladná a má vysokou kapacitu. Pro data bývá aplikována paměť typu RWM. Další plus tvoří rychlost vykonávání instrukcí, protože instrukci i data lze číst najednou (obr. 3.1, str. 17). Hlavní nevýhodou Harvardské architektury jsou dvě sběrnice, které mají vyšší nároky při vývoji řídící jednotky procesoru. To se projeví při celkových nákladech na výrobu mikročipu. Dalším menším nedostatkem je fakt, že není možné využít volnou datovou paměť pro program a naopak [3].

- Aritmeticko-logická jednotka - základní komponenta, ve které se provádějí všechny aritmetické a logické operace např.: sčítání a odčítání celých čísel, logický součin, logický součet, funkce NOT, NAND, NOR, rotace bitu a další,
- datová paměť - paměť pro data,
- programová paměť - paměť pro program,
- kontrolní jednotka - komponenta CPU, která řídí operace procesoru, a také koordinaci a komunikaci mezi vstupními nebo výstupními zařízeními,
- I/O - vstup nebo výstup dat.



Obrázek 3.1: Obecné schéma Harvardské architektury

3.1.2 RISC

RISC označuje procesory s redukovatelnou instrukční sadou. Mají malý počet instrukcí (instrukční sada tvoří 32 instrukcí). Instrukce jsou velmi jednoduché a většinou se provádějí v jednom strojovém cyklu. Neexistují žádné specializované registry, takže většina instrukcí pracuje přímo s vnitřními registry, což je daleko rychlejší. Implementace instrukcí se provádí logickými obvody. Délka, co se týče počtu bitů, je u všech instrukcí stejná. RISC umí také pipelining neboli zřetězené zpracování (ukázka rozdílů v tabulkách 3.1 a 3.2, str. 17 a 18). Jde o rozložení jedné instrukce do několika fází. Pro příklad lze v první fázi načítat a dekódovat instrukci, dále ji pak v druhé fázi provádět. Hlavní myšlenka spočívá ve využití částí procesoru, které zrovna nic nevykonávají. Tím se rapidně zvyšuje výkon celého procesoru.

1.takt	2.takt	3.takt	4.takt
1.instrukce: načtení	1.instrukce: provedení		
	2.instrukce: načtení	2.instrukce: provedení	
		3.instrukce: načtení	3.instrukce: načtení

Tabulka 3.1: Ukázka s pipeliningem

1.takt	2.takt	3.takt	4.takt
1.instrukce: načtení a dekód.	1.instrukce: provedení		
		2.instrukce: načtení a dekód.	2.instrukce: provedení

Tabulka 3.2: Ukázka bez pipeliningu

3.1.3 Základní rodiny AVR

Mikrokontroléry AVR se dělí do několika podskupin neboli rodin, a to podle periferií, počtu pinů, velikosti paměti atd.

1. **AVR - AT90 série**
   - tato řada se dnes již nevyrábí
2. **tinyAVR - ATtiny série**
   - paměť pro program: 0,5 - 16 kB
   - počet pinů: 6 - 32
   - frekvence: do 20 MHz
   - omezená sada periferií
3. **megaAVR - ATmega série**
   - paměť pro program: 4 - 512 kB
   - počet pinů: 28 - 100
   - frekvence: do 20 MHz
   - rozšířenější sada instrukcí
   - rozšířená sada periferií
   - picoPower - úspora energie, prodloužení životnosit baterie
4. **XMEGA - ATxmega série**
   - paměť pro program: 16 - 384 kB
   - počet pinů: 32 - 100
   - frekvence: do 32 MHz
   - DMA - přímý přístup do paměti (Direct Acess Memory)
   - Event System - umožňuje komunikaci mezi periferiemi bez nutnosti zatěžování CPU
   - picoPower - 2. generace

5. AVR32 - AT32UC3 série

- paměť pro program: 16 - 512 kB
- počet pinů: 48 - 144
- frekvence: do 66 MHz
- periferie podobné jako u ATxmega série

Mezi poslední patří tak trochu specifická podskupina AutomotiveAVR. Mikročipy v této skupině jsou speciálně vyráběny pro použití v automobilovém průmyslu. Vyskytují se zde upravené klony z jednotlivých rodin např. ATmega88 Automotive atd. Jejich rozdíl spočívá kupříkladu v teplotním rozsahu nebo v počtu přemazání flash paměti.

3.2 Výběr mikrokontroléru

Pří výběru jednočipového počítače hrálo roli několik důležitých aspektů. V první řadě to byla předem stanovená řada AVR. Další roli hrál také počet pinů. Modul totiž musel obsáhnout všechny komponenty výukového přípravku. Svoji roli hrála také velikost mikrokontroléru tzn. velikost pouzdra tak, aby zabral co nejmenší část modulu. Velký význam hrálo také napájecí napětí, které mělo být +5V. Za poslední důležitý bod bylo považováno možné použití Arduino platformy do jednočipového počítače. Po zvážení několika variant padla volba na MCU z rodiny megaAVR, a to konkrétně na ATmega2560-16AU [5].

	ATmega2560
Flash	256 kB
EEPROM	4 kB
RAM	8 kB
Celkový počet pinů	100
Univerzální I/O piny	86
16-ti bitové PWM kanály	12
UART	4
ADC kanály	16
Napájecí napětí	4,5 - 5,5 V
Frekvence	0-16 MHz
Pouzdro	TQFP100

Tabulka 3.3: Parametry mikrokontroléru

4. Programátor

Programátor je zařízení, které transportuje předem upravený program v hexadecimálním formátu do příslušného mikrokontroléru.

Pro jednoduché použití modulu při výuce byl kladen důraz na umístění programátoru přímo na plošném spoji. Vybraný programátor by pak komunikoval s počítačem pomocí USB. Po konzultaci s odborníky byly vybrány pro testování a porovnání dva programátory - Ponyser a USBasp.

4.1 Rozhraní ISP

USBasp i Ponyser komunikují s mikročipem pomocí ISP rozhraní (in system programming). Jedná se o takové rozhraní, které umožňuje programovat interní paměti jednočipového počítače přímo v cílové aplikaci. Toto řešení výrazně urychluje proces vývoje a předchází tak mechanickému poškození při manipulaci mezi speciálním programátorem a cílovou aplikací. Mezi další možnosti naprogramování mikročipu patří rozhraní JTAG, BOOTLOADER a další.

Existují dvě varianty konektorového zapojení ISP rozhraní, a to se šesti vodiči nebo s deseti vodiči (viz obr. 4.1 a tab. 4.1, str. 21).

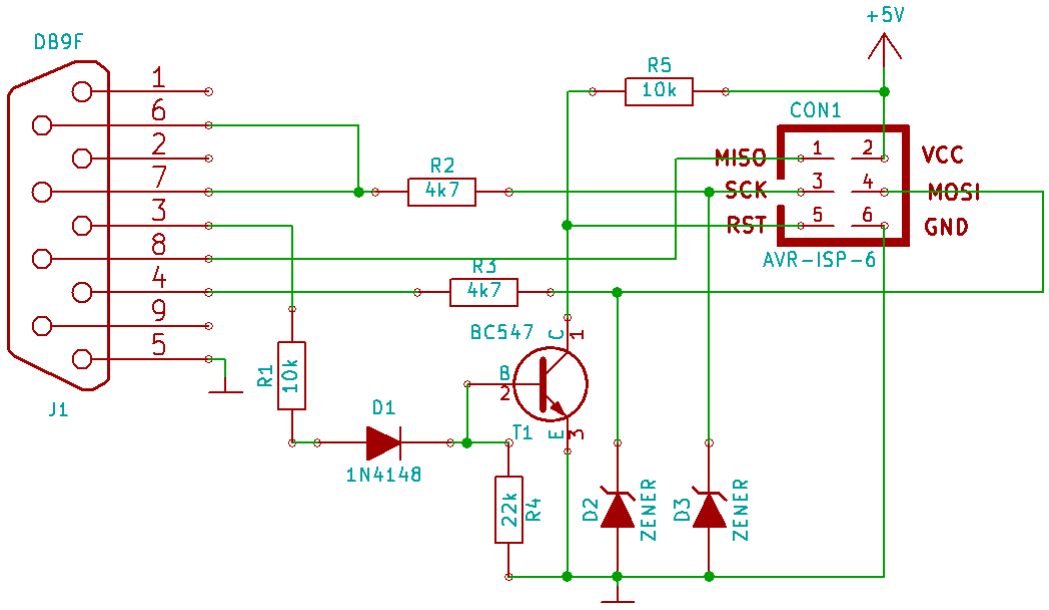
Obrázek 4.1: Konektory ISP

Signál	6-pinů	10-pinů	I/O	Význam
VCC	2	2	–	Napájecí napětí
GND	6	(3),4,6,8,10	–	Uzemnění
MOSI	4	1	Výstup	Příkazy a data z programátoru do MCU
MISO	1	9	Vstup	Data z MCU do programátoru
SCK	3	7	Výstup	Hodinový signál řízený programátorem
RESET	5	5	Výstup	Reset řízený programátorem

Tabulka 4.1: Piny rozhraní ISP

4.2 Ponyser

Tento AVR programátor patří mezi nejjednodušší a nejlevnější zařízení. Pro výrobu programátoru postačí několik rezistorů, tři diody, tranzistor a dva konektory (viz obr. 4.2) nebo [9]. Pracuje při napájecím napětí +5 V. Ke komunikaci s počítačem, který nevlastní DB9 konektor, musí být použit příslušná redukce.



Obrázek 4.2: Schéma programátoru Ponyser

4.3 USBasp

Toto AVR zařízení používá pro naprogramování mikrokontrolér integrovaný obvod. Zpravidla to bývá mikročip z rodiny megaAVR, který obsahuje určitý firmware. Pro připojení k PC používá USB konektor. Napájecí napětí programátoru

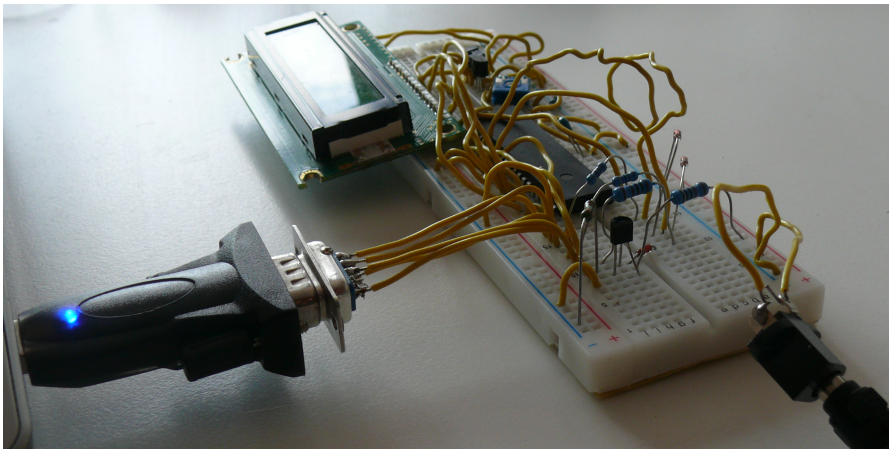
mohou být dvě, a to +3,3 V nebo +5 V. Umožňuje tři důležitá nastavení pomocí propojení určitých vodičů (viz příloha C obrázek 2):

- přepnutí napájení mezi USB nebo externím zdrojem napětí (konektor - JP1),
- programování mikrokontrolérů, které mají hodinovou frekvenci menší než 1,5 MHz (konektor - JP2),
- přeprogramování samotného integrovaného mikročipu na programátoru novým nebo jiným firmwarem (konektor - JP3).

Obsahuje také dvě LED diody, které signalizují napájení a přenos programu do mikročipu. Pro vyšší hodinovou frekvenci MCU na programátoru se používá externí krystal s hodnotou 12MHz.

4.4 Testování

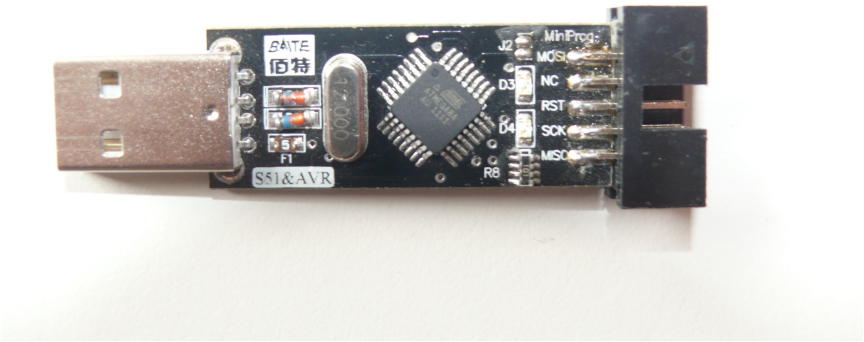
Jelikož přenosová rychlost programátoru Ponyser nebyla určena, odvodila se z přenosu vybraných kódů vzestupně podle velikosti do mikročipu v nepájivém poli (v tomto případě Atmega16). K propojení počítače s programátorem byl použit převodník USB - RS-232 (viz obr. 4.3).



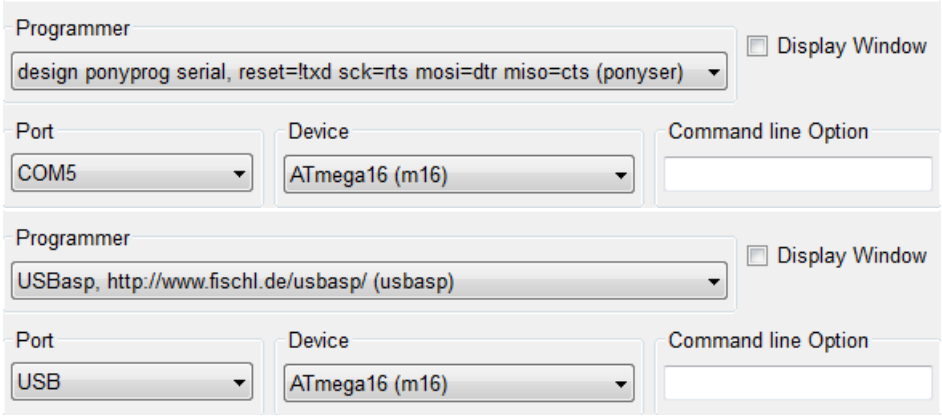
Obrázek 4.3: Ponyser na nepájivém poli

Pro lepší srovnání byl také testován programátor USBasp (obr. 4.4, str. 23), který nebyl sestaven nepájivém poli, ale přímo koupen v internetovém obchodu. Firmware byl v mikrokontroléru předem nahraný. V softwaru avrdude-GUI (kapitola 6.1.1) se pak nastavily příslušné hodnoty pro jednotlivé programátory (viz obr. 4.5, str. 23).

Poté se začalo s nahráváním jednotlivých kódů vzestupně podle velikosti. Programátor Ponyser začal postupně ztrácet. Program, který činil 2141 B, nahrával 33



Obrázek 4.4: Testovací USBasp



Obrázek 4.5: Nastavení programátorů v avrdude-GUI

sekund. USBasp to dokázal do 2 sekund. Testování rychlosti tedy jednoznačně rozhodlo pro USBasp. Nakonec byly shrnuty veškeré výhody i nevýhody jednotlivých programátorů.

Jako použitelnější programátor pro umístění na modul byl finálně vybrán USBasp, který umožňuje větší přenosovou rychlost i schopnost přímého propojení s počítačem pomocí USB konektoru. Další velikou výhodou tohoto programátoru je možnost aktualizace firmwaru.

5. Návrh desky plošného spoje

Před samotným návrhem došlo k pečlivému vybírání softwaru, který by dostatečně uspokojil všechny předem určené požadavky. Jednalo se především o cenu programu, množství vrstev DPS, maximální velikost desky plošného spoje, počet možných komponent, možnost vytvoření vlastní knihovny součástek nebo přidání existující a v poslední řadě také internetová podpora programu. Přednost před programem Eagle dostal nakonec software KiCad.

5.1 O programu KiCad

Tento program byl vytvořen v roce 1992 Jean-Pierre Charrasem [12]. Na jeho dalších verzích se nyní podílí už více lidí tzv. KiCad Developers Team. Software je volně šiřitelný s licencí GNU GPL v2. Mezi jeho hlavní výhody patří podpora pro Windows, Linux i Apple OS X, dále možnost neomezené velikosti schématu a plošného spoje (může mít až 16 vrstev) a široká základna knihoven na internetu, které mohou vkládat také samotní uživatelé. Je zde možnost trojrozměrného návrhu součástek, ale i 3D pohled na aktuální plošný spoj. Pro program Kicad je k dispozici také čeština. Bohužel některá slova nebyla z angličtiny přeložena.

5.2 Popis programu Kicad

5.2.1 Editor schématu

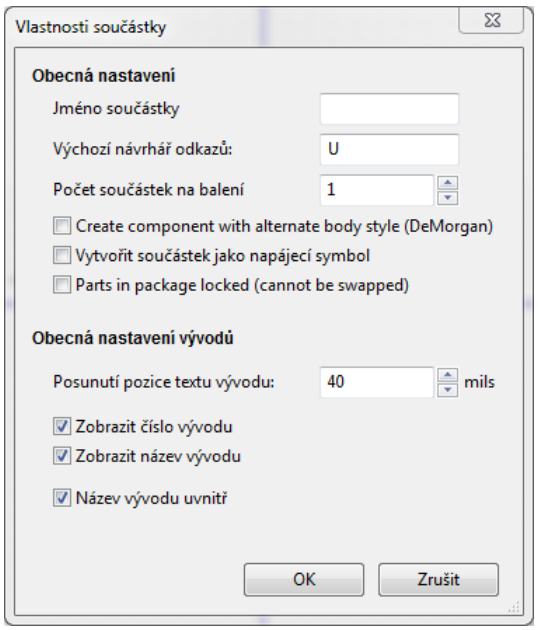
První integrovaný software Kicadu nese název Eeschema. Umožňuje navrhnout schéma elektrického obvodu. Má na výběr tři druhy: jednostránkové, vícestránkové a hierarchické. Dále nabízí umísťování součástek a napájecích portů, propojovací spoje, sběrnice i jejich textový popis, označení nezapojených spojů, přidávání propojení, kopírování vybraných bloků, výběr mřížky v palcích nebo milimetrech.

- **Prohlížeč knihoven**

Zobrazuje jednotlivé součástky v určených knihovnách. Používá se hlavně pro celkový přehled i výběr knihoven, které budou ve schématu použity. Lze také zobrazit jednotlivé datasheety součástek ve formátu PDF.

- **Editor knihoven**

Tento program umožňuje vytváření nových knihoven i součástek (obr. 5.1). Uživatel má na výběr několik možností: vytvořit součástku do nové nebo předem vybrané knihovny, načíst ji ze stávající knihovny a upravit její vlastnosti, vytvořit novou součástku ze stávající.



Obrázek 5.1: Vytvoření nové součástky

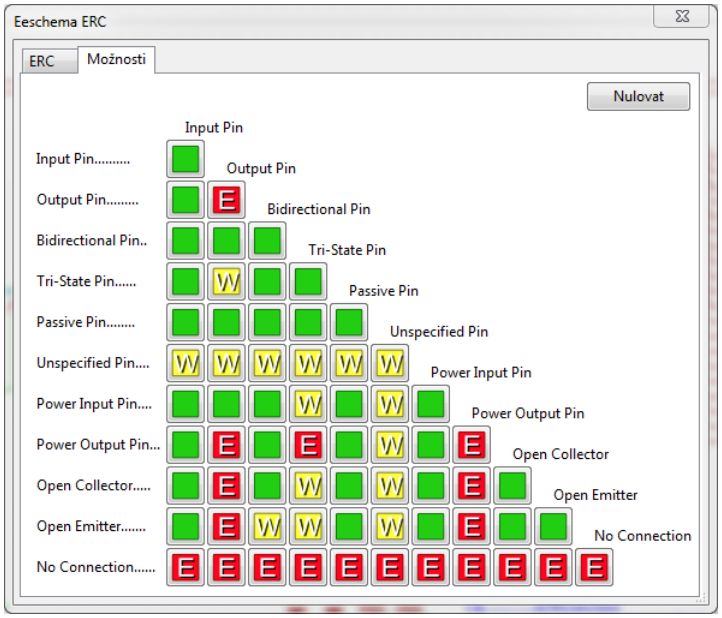
Následuje vytváření grafické stránky, která umožňuje vykreslit schématickou značku součástky. Při vložení pinu součástky lze nastavit spoustu parametrů jako je např. číslo, druh vývodu, orientace atp.

- **Očíslování součástek ve schématu**

Každou komponentu v návrhu je nutno očíslovat, kvůli generování netlistu. V tomto programu se dá nastavit několik kritérií. Prvním z nich je rozsah, který nabízí očíslovat aktuální stránku nebo celou hierarchii. Další nabízí smazat aktuální číslování nebo ho ponechat a doplnit, značit součástky podle pozice hodnoty X a Y. Lze také nastavit výběr očíslování od určitých hodnot: použije první volné číslo ve schématu, začne od čísla listu schématu vynásobeného 100 nebo 1000 a použije první volné číslo.

- **Kontrola elektrických pravidel**

Kontrola elektrických pravidel (v programu někdy pod názvem ERC) hlídá chyby uživatele způsobené při návrhu schématu (např. zapojení výstupu na výstup atd.). Vyskytuje se také v nabídce při tvoření nové součástky, kde plní stejnou funkci. Veškeré nastavení těchto pravidel nabízí kontrolní matice(obr. 5.2, str. 26). Uživatel



Obrázek 5.2: Kontrolní matice

si sám zvolí, které chyby má ERC hlídat. Existují tři možnosti. Při propojení dvou signálů program nahlásí varování, error nebo na nic neupozorní tzn., že je vše v pořádku.

- **Generování netlistu**

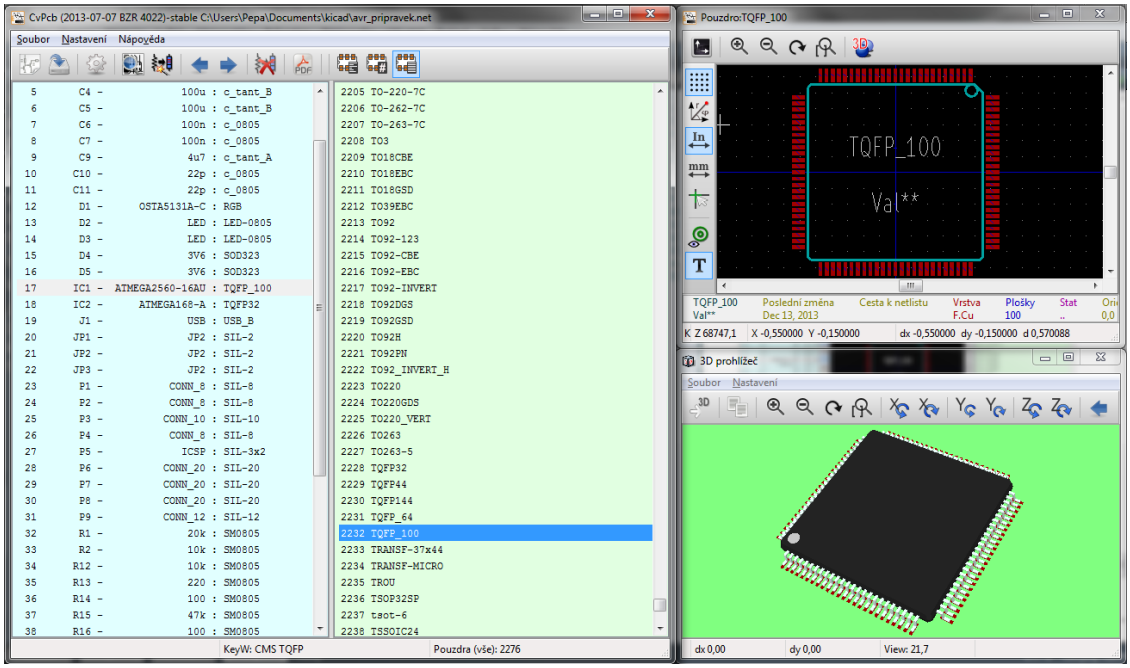
Tento program umožňuje přechod do návrhu DPS generováním netlistu (seznam součástek a spojů). Kicad umožňuje export i do jiných návrhových programů. Například OrCad, CadStar, Spice. Dají se také použít různé zásuvné moduly, takže export může být ještě více rozšířen.

- **Generování rozpisu materiálu**

Jak už sám název napovídá, tento program dokáže vygenerovat seznam všech součástek, jejich hodnoty i pouzdra. Toho se využívá především při tvorbě objednávek, kdy program nabídne uživateli hotový soupis komponent. Seznam se dá otevřít téměř v každém textovém editoru.

5.2.2 Přiřazování pouzder

Většina součástek se vyrábí v několika pouzdrových variantách. Požadované přiřazení v programu KiCad řeší software s názvem CvPcb. Součástka ve schématu získá určité rozmístění pájecích plošek, které je nutné k návrhu desky plošného spoje. Stejně jako existuje knihovna součástek pro návrh schématu, existuje také knihovna pouzder. Uživatel v programu nahraje generovaný netlist, a tím se následně zobrazí veškeré používané součástky ve schématu. Po nahrání knihovny pouzder lze propojit součástku s vybraným pouzdrem. Poté je vše připraveno k návrhu plošného spoje.



Obrázek 5.3: Ukázka softwaru přiřazení pouzder

- **Zobrazení vybraných pouzder**

V programu existuje také funkce označovaná jako *Zobrazit vybrané pouzdro*, která uživateli nabídne celkový vzhled součástky a rozmístění pájecích plošek. Umožňuje i 3D pohled (viz obr. 5.3).

5.2.3 Editor plošného spoje

Třetí program z nabídky KiCad má název Pcbnew. Vytváří konečný návrh desky plošného spoje. Mezi jeho hlavní funkce patří vytváření spojů a průchodek, dodatečné přidání modulů, vyplnění vodivými oblastmi nebo zvýraznění sítě vybraného signálu.

V dalších podkapitolách budou vysvětleny další podprogramy a důležité funkce, které jsou potřeba pro zhotovení plošného spoje.

- **Nahrání netlistu**

Při vytvoření nové DPS se v programu nejprve nahraje netlist, který si uživatel sám vybral a nastavil v předchozích programech. To umožní zobrazit jednotlivé komponenty propojené signálovými čarami tak, jak byly ve schématu navrženy.

- **Rozmístění součástek**

Při importu součástek do programu se stává, že dojde k překrývání jednotlivých komponent. Kvůli celkovému přehledu existuje v programu funkce *Režim pouzdra*. Ta umožňuje automatické přesouvání součástek, což je potřeba pro přehled veškerých komponent a jejich další nastavení.

- **Technické vrstvy**

Návrh DPS je rozdělen do několika vrstev. Každá plní svou vlastní roli. Rozdělení významných vrstev v programu KiCad je následující:

Cu	- vrstva mědi
Adhes	- vrstva lepidla (pro SMD součástky)
Paste	- vrstva pájecí pasty
Silks	- vrstva rozmístění potisku
Mask	- vrstva nepájivé masky
User	- uživatelská vrstva
Cuts	- rozměr desky plošného spoje

- **Module Editor**

Tento podprogram slouží k vytváření nebo upravování jednotlivých pouzder. Před samotným návrhem se vybere tzv. aktuální knihovna, do které má být toto pouzdro umístěno. Po zadání názvu může začít samotný návrh. Dají se vkládat různé druhy a velikosti padů. Aby bylo zaručeno správné propojení symbolu s pouzdrem součástky na plošném spoji, musí mít piny i pady stejná čísla. Existuje zde také možnost ohraničení součástky pomocí čáry, mnohoúhelníku atd. K vytvořenému pouzdru lze pak přiřadit i 3D model z programu Wings3D.

- **Autoroutery**

Při složitém návrhu plošného spoje lze využít program tzv. autorouter, který sám navrhne jednotlivé vodivé cesty. Uživatel má na výběr hned dva. První se nachází přímo v editoru DPS. Pro správnou funkčnost se musí nastavit návrhová pravidla přímo v PCB editoru. Druhý, s názvem FreeROUTE, je k dispozici přes internet pomocí Java pluginu. Obsahuje vlastní nastavení návrhových pravidel.

- **Vykreslení**

Pro vyrobení plošného spoje je nutno vykreslit jednotlivé vrstvy, které budou dále poslány do výroby. Volbou *Kreslit* v programu otevře uživatel menu, ve kterém vybere příslušné vrstvy a soubor vrtání. Poté vybere formát: Gerber, Postscript, SVG, DXF nebo HPGL.

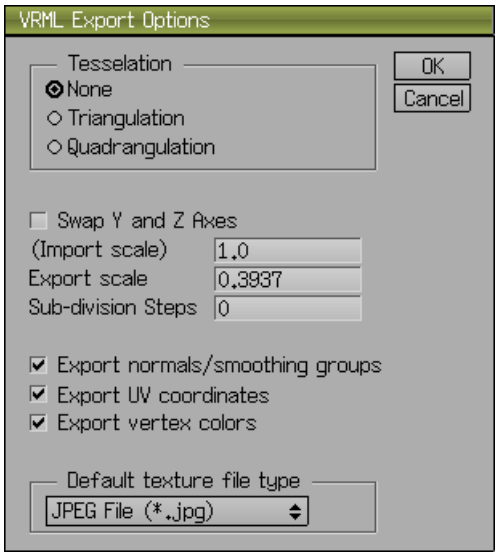
5.2.4 Prohlížeč Gerber

Čtvrtý podprogram v nabídce Kicad, který byl v bakalářské práci použit se nazývá GerbView. Umožňuje uživateli prohlížet a tisknout soubory Gerber i vrtací soubory Excellon.

5.2.5 3D modeling

Externí program pro modeling součástek ve trojrozměrném prostoru umožňuje program, který má název Wings 3D. Je velice snadný a spolehlivý. Vyznačuje se intuitivním ovládáním, což každý začátečník ocení. Hodí se například pro spolupráci mezi návrhářem a konstruktérem.

Některé součástky AVR modulu neobsahovaly 3D pohled. Byly tedy navrženy ve zmíněném softwaru a exportovány do programu KiCad. Pro správnou velikost součástky v 3D vizualizaci návrhového softwaru musel být přesně nastaven exportní rozměr (viz obr. 5.4). Celkový pohled na výsledný 3D AVR modul (viz příloha B obrázek 1).



Obrázek 5.4: Exportní nastavení součástky

5.3 Postup při návrhu schématu

Po výběru MCU a vhodného programátoru bylo možno přejít k samotnému návrhu schématu. Nejprve došlo k převzetí a upravení (záměnou mikročipu) programovací části schématu (viz příloha C obrázek 8). Pro možnost resetovat řídící mikročip bylo na resetovací pin připojeno tlačítko. Vložením externího krystalu mezi určité piny se zajistila přesná hodinová frekvence řídícího integrovaného obvodu ATmega2560. LED bargraf na přípravku se vždy na ostatních výukových modulech obsluhoval pomocí dvou 8-bitových posuvných registrů. Byly tedy také zařazeny do schématu (viz příloha C obrázek 3). Pro možnost připojení zajímavých komponent od firmy Arduino byly zahrnuty také konektory pro Arduino Shield. Po propojení signálů zbylo několik volných pinů mikrokontrolér ATmega2560. Vybrali se proto ještě další tři komponenty, které se vložily do schématu modulu, a to RGB LED,

optické čidlo a senzor světla. Ostatní nepoužité piny byly signály vyvedeny na samostatný konektor pro případné použití.

Při navrhování schématu se některé symboly museli navrhovat vlastnoručně (viz kapitola 5.2.1). Po finálním návrhu signálových cest se komponenty očíslovaly a proběhla kontrola elektrických pravidel. Nakonec bylo výsledné schéma použito pro generování netlistu.

5.4 Postup při výběru pouzder

Při výběru pouzder jednotlivých komponent se kladl důraz na co nejmenší rozměry, protože velikost modulu nebyla zrovna největší. Spolu s tímto faktorem hrála důležitou roli také možnost pájení součástek pomocí přímého pájecího hrotu s průměrem špičky 0,5 mm. Jako nejmenší SMD pouzdro bylo tedy vybráno 0805. V poslední řadě se bral ohled i na dostupnost součástek v internetovém obchodu.

Některá pouzdra bohužel nebyla v nabídce programu CvPcb. Vytvořila se tedy ručně v programu Module Editor.

5.5 Postup při návrhu DPS

V návrhu desky plošného spoje se muselo dodržet umístění všech součástek na horní straně oboustranné DPS. Jedinou výjimku tvořily konektory, které se zasunovaly do výukového přípravku. Mělo to pomoci k lepšímu vysvětlení významu jednotlivých komponent při výuce. Další výhodou byl komfortnější přístup při hledání jakékoliv závady na plošném spoji (viz příloha C obrázek 5, 6 a 7).

První věcí v programu Pcbnew bylo vyznačení obrysu modulu, který se přejal z layoutu starších modulů používaných při výuce. Rozmístění konektorů, jejichž výstupy ovládaly komponenty přípravku, byla také převzata ze stejného zdroje. Konektory pro Arduino Shield byly rozestavěny podle internetového schématu pod licencí GNU GPL na stránkách firmy spolu s ISP rozhraním [14]. Poté došlo k umísťování integrovaných obvodů ATmega2560, ATmega168 i jejich externích krystalů, posuvných registrů a součástek ve schematické části USBasp. Nakonec byly rozestavěny tyto komponenty: reset tlačítko, senzor světla [17], RGB LED [16] a optický senzor [18] tak, aby nebyly po zasunutí libovolného Arduino Shieldu zakryty.

Po rozmístění všech součástek následovalo jejich propojení signálovými i napájecími vodiči. V nastavení se zvolila jednotná šířka vodiče, která byla schopna připojit sousední vodiče k pouzdru TQFP100 s dostatečnou mezerou. Předně se začaly vést signály napěťové a zemnící. Pak nastalo propojování signálů ISP rozhraní. Následovalo vedení cest od programátoru (ATmega168 [6]), ale i od komponent výukového přípravku k mikročipu ATmega2560. Při tak vysokém počtu cest na malém prostoru muselo být použito několik desítek průchodek.

6. Programování modulu

Pro demonstraci funkčnosti vyrobeného AVR modulu (viz příloha B obrázek 8) bylo nutno naprogramovat vhodné příklady. Byly sestaveny dva programy. První se naprogramoval v Atmel Studiu v jazyku C. Druhý byl sestaven v prostředí Arduino. Nejprve se však musel nastavit programátor.

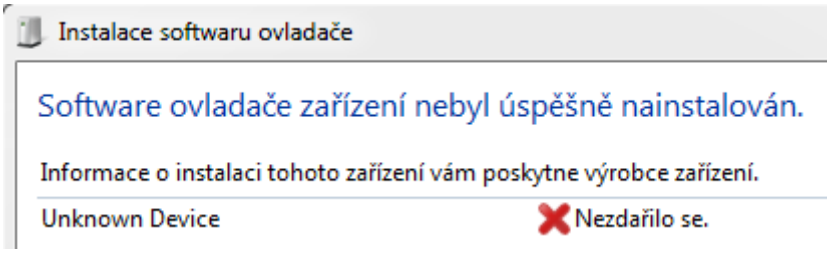
6.1 Naprogramování USBasp

První věcí po zhotovení a osazení modulu bylo nutno naprogramovat příslušný integrovaný obvod (ATmega168) tak, aby se choval jako programátor. Firmware byl stažen z internetového zdroje [8] pod licencí GNU GPLv2 přímo v hexadecimálním tvaru. Pro nahrání kódu do integrovaného obvodu byl použit externí programátor AVRProg USBv2 [23] (podobná vnitřní struktura jako u USBasp). Protože tento programátor obsahoval 10-ti pinový konektor ISP, musela být vyrobena redukce, aby došlo ke spojení příslušných pinů v modulu (viz příloha B obrázek 9). Reset pin však nebyl připojen na ISP rozhraní, ale na konektor JP2 pin 1. Následovalo propojení konektorů pomocí jumperů na plošném spoji modulu, a to JP1 a JP3 (viz obr. C.2). Poslední věc, která byla nutná ke správnému fungování firmwaru v MCU, bylo nastavení jednotlivých interních propojek. Na internetovém kalkulátoru [21] byly nastaveny určité parametry, které definovaly výslednou hodnotu propojek (viz obr. B.2). V softwaru avrdude-GUI byly nastaveny jednotlivé parametry a cílová cesta k hexadecimálnímu tvaru firmwaru (viz příloha B obrázek 4). Po stisknutí tlačítka *Write* v rámečku *Fuse* došlo k přepsání propojek. Konečnou programovací fází bylo kliknutí na položku *Write* v rámečku *Flash*. Po odpojení externího programátoru a odebrání propojky JP1 na plošném spoji bylo potřeba nainstalovat ovladač pro správnou funkci USBasp programátoru. USB konektor na AVR modulu se propojil pomocí redukčního kabelu k PC. Objevila se informace (viz obr. 6.1, str. 32).

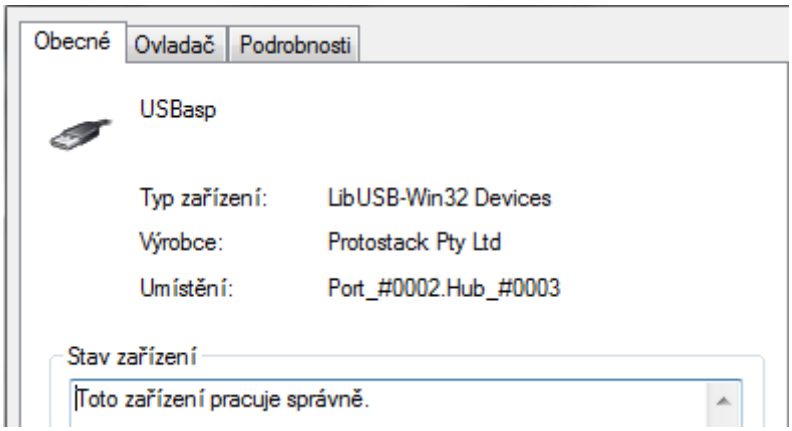
Ve vlastnostech USB portu bylo potřeba vybrat správný ovladač. Ten obsahoval soubor ve formátu ZIP, ve kterém se nacházel také firmware. Po aktualizaci se port přihlásil jako USBasp (viz obr. 6.2, str. 32).

6.1.1 AVRDUDE

Tento program umožňuje nahrávat nebo stahovat programy z paměti Flash a EEPROM z AVR mikrokontroléru pomocí ISP. Název toho nástroje vznik z anglických slov (AVR Downloader/UploaDEr). Byl napsán Brianem S. Deanem, který



Obrázek 6.1: První připojení USBasp



Obrázek 6.2: Nainstalovaný ovladač USBasp

ho prvně nazval AVRPROG a prvně fungoval na operačním systému FreeBSD. Kvůli velikému zájmu vznikla i podpora pro Windows. Aby se vyhnul konfliktu s AVRPROG.EXE (Atmel's Windows programming software), přejmenoval název na AVRDUDE. Dnes tento program podporuje také Linux i Apple OS X a patří do licenční skupiny GNU GPL v2 [20].

Spuštění toho programu se provádí v příkazovém řádku pomocí slova *avrdude*. Pomocí jednotlivých příkazů nastavíme potřebné parametry.

Pro lepší a přehlednější programování lze použít avrdude-GUI. Tento soubor umožňuje grafické uživatelské rozhraní (zkratka GUI viz příloha B obrázek 5). Obsahuje výběr z několika desítek mikrokontrolérů a druhů programátorů. Při *Device* a *Programmer*. Po určení těchto parametrů následuje výběr portu *Port*, na kterém se programovací zařízení nachází. Dále je zde možno určit vnitřní nastavení mikročipu pomocí interních propojek *Fuses*. V prostření avrdude-GUI lze jednoduše vybírat soubory, které se mají nahrát do paměti Flash nebo EEPROM. Jednotlivé paměti se dají číst i přepisovat. Existuje zde také tlačítko *Terminal*, které zjistí dostupnost MCU po předem určených parametrech (tj. zvolený programátor, port a zařízení). V případě neshody nebo nesprávného připojení programátoru k mikročipu náležitě upozorní vhodným komentářem. Označení *Lock Bit* slouží k uzamčení firmwaru. Tlačítko *Chip Erase* vymaže paměť Flash i EEPROM v jednočipovém počítači. Posledním nástrojem v programu je *Command line Option*, který je totožný s pro-

gramováním mikrokontroléru v příkazové řádce.

6.1.2 Programovací propojky (Fuse Bits)

Většina mikročipů lze speciálně nastavit pomocí tzv. programovatelných propojek [5]. Slouží hlavně k nastavení hodinového signálu MCU, ochraně proti přepisu atd. Mezi nejzákladnější patří:

Propojka	Význam
CKDIV8	dělička frekvence hodin osmi
CKOUT	hodinový výstup na továrně určený pin
SUT[1..0]	doba pro spuštění MCU (určená krystalem)
CKSEL[3..0]	určení frekvence hodin: interní,externí atd.
SPIEN	komunikace přes ISP
WDTON	při zacyklení resetuje procesor
EESAVE	EEPROM paměť je při přemazání zachována
BODLEVEL[2..0]	reset mikročipu při poklesu napájecího napětí

Tabulka 6.1: Propojky

6.2 První úloha

V zadání bakalářské práce nebyly nikde přesně stanoveny příklady, které by měly výsledný modul otestovat na výukovém přípravku. Byl tedy vymyšlen program v jazyku C, který demonstroval použití komponent přípravku i modulu.

6.2.1 Atmel Studio

Veškerý kód byl sepsán v programovacím prostředí Atmel Studio 6.1 [19]. Přednost dostalo před WinAVR díky inteligentnímu asistentovi doplňování kódu. Tato funkce predikuje uživatelův záměr při psaní instrukce a nabízí mu finální varianty. Není proto nutné časté nahlížení do datasheetu součástky. Někdy se tato funkce vyskytuje pod pojmem IntelliSense.

Mezi hlavní parametry programu patří:

- podpora pro 8-bitové, 32-bitové AVR mikrokontroléry a další například ARM,
- programovací jazyky: C/C++ a assembler,
- integrovaný kompilátor a debugger,
- integrované základní knihovny,
- již zmíněná asistence doplňování kódu.

6.2.2 Funkce programu první úlohy

Po přivedení napájecího napětí začnou LED diody a žárovka svítit, LED bargraf blikat, piezosirénka bzučet, RGB LED dioda postupně střídat tři základní barvy a na LCD displej je vyslán nápis *Vítá Vás přípravek TUL*. Po pěti vteřinách dojde k smazání displeje a k přejití do módu menu. Na displej se postupně začne psát dvojice za sebou jdoucích nápisů, každý na jeden řádek, které reprezentují jednotlivé módy. Tlačítky na přípravku lze posouvat jednotlivé položky nebo vstupovat i vystupovat z vybraných režimů. Seznam všech položek v menu je následující:

- Reálný čas

Na výukovém přípravku je umístěn integrovaný obvod DS1338, obvod reálného času RTC [11], který počítá sekundy, minuty, hodiny, měsíce, den v týdnu a rok. Obsahuje integrovaný krystal s hodnotou 32 768 Hz, který přesně počítá dané cykly. Pomocí sériového rozhraní I^2C komunikuje s MCU rychlostí 100 kHz.

Při programování tohoto prvního režimu byly použity komunikační TWI knihovny, které byly upraveny přímo na tento mód. V první řadě se mikrokontrolér dotázal na určenou adresu RTC obvodu. Když hodiny reálného času dokázaly právně odpovědět, navázala se komunikace. Poté se nahrály příslušné hodnoty sekund, minut atd. do registrů RTC obvodu. Při každém zobrazení tohoto módu byl jednoduše vyžádán aktuální čas od RTC obvodu. Po odpojení napájecího napětí nebyly údaje o čase ztraceny díky záložní baterii umístěné na přípravku.

- Senzor teploty

Druhá položka v menu zobrazuje aktuální teplotu v prostředí výukového přípravku [15]. Výstup tohoto čidla generuje obdélníkový signál s frekvencí 1-4 kHz. Změna střídy se mění v závislosti na teplotě. Teplotní čidlo bylo připojeno k pinu mikrokontroléru, který umožňoval přerušení.

Při psaní programovacího kódu pro zjištění teploty je nejprve potřeba nastavit přerušení pro každou hranu signálu. Dále pak povolit 16-bitový časovač pro určení doby signálu v jednotlivém stavu. Z těchto hodnot se určila střída signálu (viz obr.6.3). Následoval výpočet teploty. Po zkonvertování dat se hodnota teploty

```
itoa(hodnota_timeru1,displej,10);
itoa(hodnota_timeru0,displej,10);
LCDgoto(0,1);
cli();// zastavení možnosti přerušení pro jistý výpočet teploty
strida=((hodnota_timeru1)/(hodnota_timeru1+hodnota_timeru0));//určení střídy
teplota=(strida-0.32)/(0.0047);//výpočet teploty vzorcem z datasheetu
dtostrf(teplota,6,1,char_teplota);//konverze teploty do formátu k výpisu na displej
LCDsendString(char_teplota);
LCDsendString(" ");
LCDsendChar(0b11011111);// zobrazení ° (stupně)
LCDsendString("C");
sei();// povolení přerušení
```

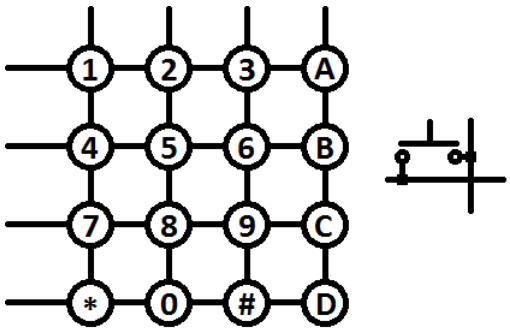
Obrázek 6.3: Ukázka výpočtu teploty

umístila na displej. Pro zahřívání teplotního čidla mohla být použita žárovka, která se rozsvítila po stisku pravého tlačítka na výukovém přípravku.

- Psaní na LCD

V tomto módu mohl uživatel zobrazovat číslice na LCD displej po stisknutí klávesy z maticové klávesnice (obr. 6.4). Byla zde možnost zapisovat jednotlivá čísla od 0 do 9 a dva znaky * a # v libovolném pořadí. Při stisku klávesy A dojde k přemístění kurzoru na první pozici prvního řádku LCD displeje. Obdobně i klávesa B způsobí přesun kurzoru, v tomto případně však na první pozici 2. řádku. Klávesa C provede vymazání veškerých číslic a nastaví kurzor na první pozici prvního řádku. Pomocí poslední klávesy D lze přepnout kurzor. Může buď blikat nebo být vypnut.

Celý program funguje na principu změny logických úrovní jednotlivých pinů, které jsou připojeny ke klávesnici. Čtyři piny, které jsou připojeny ke sloupcům matice, se v programu nastavily jako výstupy. Další čtyři piny byly připojeny k řádkům matice a nastaveny jako vstupy. Protože přípravek obsahuje zvedací rezistory, byl stav vstupů v pozitivní logice logická "1". Poté následoval samotný algoritmus na vyhledávání klávesy. Šlo o postupnou změnu jednoho výstupu (sloupce) z logické "1"do logické "0". Ostatní tři výstupní piny byly nastaveny do logické "1". Dále program zkontroloval hodnoty vstupů (řádků). Pakliže došlo ke změně vstupu do logické "0", muselo nastat propojení (stisk klávesy) aktuálně kontrolovaného řádku s aktuálním sloupcem s hodnotou v logické "0". Tento algoritmus byl nadále použit pro kontrolu ostatních sloupců. Vždy se tedy nastavila logická "0"do jiného výstupního pinu. Jestliže nenastal stisk klávesy, byla vrácena defaultní hodnota a celkový algoritmus se začal opakovat.



Obrázek 6.4: Ukázka konstrukce klávesnice

- Posuvný registr

Čtvrtá položka předvádí funkci LED bargrafu na třech naprogramovaných úlohách. První s názvem *Blikání* naráz zapíná a vypíná LED diody baru v krátkém časovém intervalu. Další *Knight Rider* simuluje jeden z efektů chytrého auta v populárním americkém seriálu Knight Rider. Tato úloha postupně zapne LED diody v bargrafu. Po zaplnění baru rozsvícenými diodami začne zhášení od poslední

rozsvícené k první. Poslední s názvem *Střídání* zapíná a vypíná jednotlivé LED diody postupně ob jednu.

Při programování těchto aplikací byla nutná znalost funkce 8-bitových posuvných registrů 74HC595 [10]. Dva totiž ovládaly všech deset diod v bargrafu pomocí pěti hlavních pinů, které byly řízeny MCU ATmega2560 (tab. 6.2). Celý algoritmus spočíval ve správném nastavení a zobrazení hodnot posuvného registru. Každá úloha nejprve začínala příslušným nápisem na LCD displej. Poté následovalo vlastní vykonávání daného algoritmu.

Signál	Vysvětlení
SER	Hodnota pro další posuv
SRCLK	Při přivedení hodinového signálu posune registr
RCLK	Ve stavu logické "1"zobrazí hodnoty registru na výstupy
SRCLK	Ve stavu logické "0"vymaže posuvný registr
$\overline{OH}$(Q7')	Výstupní hodnota přivedená do 2. regitru na pin SER

Tabulka 6.2: Hlavní piny posuvného registru

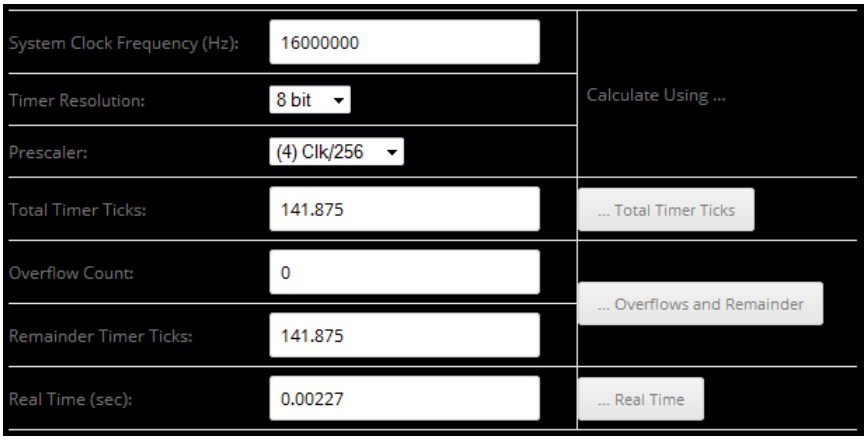
- Písničky

Pro předvedení zvukového efektu přípravku byla použita piezosirénka. Ta umožňuje přehrát dva druhy písní. První písnička *Ovčáci čtveráci* je definována v dvoučárkované oktávě tj. od c" (523,2 Hz) do h".Druhá *Skákal pes* byla naprogramována v oktávě jednočárkované tzn. od c' (261,5 Hz) do h'.

Při programování piezosirénky nešlo použít výstup z časovače, jelikož to pin procesoru neumožňoval. Většina takovýchto pinů byla totiž použita pro PWM Arduino Shieldu. Použilo se tedy přerušení 8-bitového časovače při CTC. Výpočet hodnoty jednotlivých tónů byl proveden na internetovém kalkulátoru [22] (viz obr. 6.5, str. 37), kde se nastavily příslušné hodnoty časovače i převrácená hodnota frekvence tónu podle známého vztahu. Příklad výpočtu tónu a1 (frekvence 440 Hz).

$$T = \frac{1}{f} = \frac{1}{440} \doteq 0,00227 \quad s^{-1} \tag{6.1}$$

Při dosažení časovače vypočítané hodnoty aktuálního tónu se invertoval logický stav pinu piezosirénky. Frekvence tónu byla tedy dvakrát tak veliká (tón zněl o oktávu níž). Proto se výsledná hodnota z internetového kalkulátoru musela dělit dvěma.



Obrázek 6.5: Výpočet tónu a1

- Senzor světla
- Senzor odrazu
- Potenciometr

Tyto tři módy mají mnoho společného. Hodnota výstupu je analogová, takže byly připojeny k pinům mikročipu, které umožňovaly analogově-digitální převod. Inicializace a ostatní nastavení proběhly v jedné funkci (viz obrázek 6.6). V prvním

```
void adc_init(void)
{
    ADCSRA|=(1<<ADEN);//povolední analogově - digitálního převodu
    ADCSRB|=(1<<MUX5);//povolení pinů ADC8-ADC15
    ADMUX |=(1<<REFS0);//analogové napětí je nastaveno externím
    // kapacitorem u referenčního pinu analogového pinu
    ADCSRA|=(1<<ADIE);//povolení přerušení analogově - digitálního převodu
    ADCSRA|=(1<<ADPS0)|(1<<ADPS1)|(1<<ADPS2);//16MHz -> 16000000/50000(50KHz)=320
                                              //16000000/200000(200kHz)=80
    //defaultní hodinová frekvence je od 50kHz do 200kHz
    //podle výsledků výše vybereme z tabulky v datasheetu ADPS0, ADPS1 i ADPS2
    sei();//povolení globálního přerušení
    ADCSRA|=(1<<ADSC);//start první konverze
}
```

Obrázek 6.6: Funkce inicializace ADC

z těchto tří módů byla měřena intenzita dopadajícího světla pomocí fototranzistoru KP-2012P3C [17] na AVR modulu. Při zvyšujícím se záření dochází k otvírání přechodu mezi bází a emitorem, a tím k průchodu proudu.

V druhém módu byl použit optický senzor CNY70, který fungoval na principu odrazu infračerveného světla od různých předmětů. V pouzdru tohoto čidla jsou dvě součástky, a to infračervená dioda a fototranzistor [18].

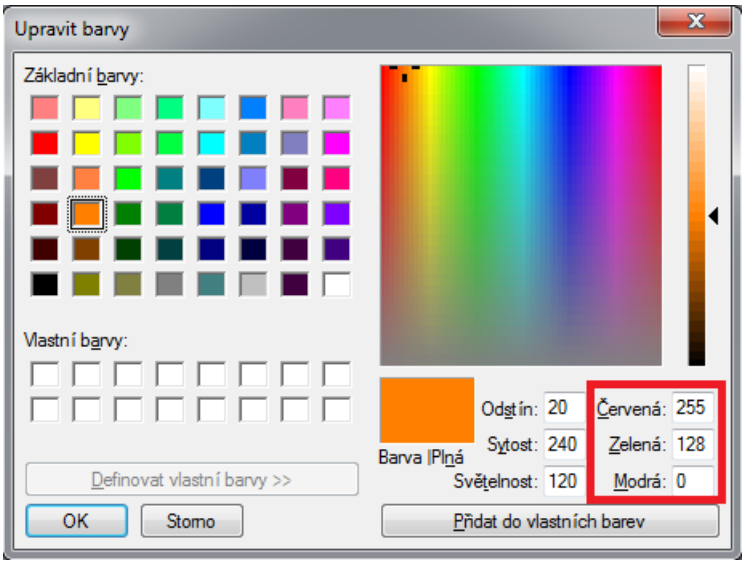
Poslední analogový mód snímal hodnotu uhlíkového mono potenciometru na výukovém přípravku. Při otáčení jezdcem se mění hodnota elektrického odporu. Tím také dochází ke změně napětí na výstupním pinu.

Hodnotu fototranzistoru a optického členu bylo vhodné zobrazovat v procentech pro lepší představivost uživatele. Proto tedy došlo k testování maximální vyčíslené hodnoty. V případě fototranzistoru byla použita žárovka (40 W), u optického členu byla pro tuto hodnotu použita deska černé barvy. Maximální výsledná hodnota se pak považovala za stoprocentní. Hodnota potenciometru byla jako jediná zobrazována v desetibitové hodnotě.

- RGB led

Poslední položka v menu patřila RGB LED diodě [16]. Při zvolení tohoto módu začal přehled několika barev v tomto pořadí: červená, oranžová, žlutá, zelená, světle modrá, modrá, růžová. Tato speciální dioda se skládá ze tří LED diod: červené, zelené a modré, které jsou dohromady umístěny v jednom skleněném pouzdře.

Každá LED dioda byla programována zvlášť pomocí přerušení při CTC módu časovače, obdobně jako u piezosirénky. V tomto případě se však jednalo o tři 16-bitové časovače. Správné hodnoty frekvence k vytvoření požadované byly brány z možnosti *Upravit barvy* v programu *Malování*. V případě oranžové barvy (viz obr. 6.7) se časovač s červenou LED diodou nastavil na maximální hodnotu tj. 65 535 a 32 768 pro hodnotu zelené barvy. Modrá LED dioda zůstala vypnutá. Obdobně se takovou metodou dá nastavit téměř jakákoliv barva.



Obrázek 6.7: Nastavení oranžové barvy

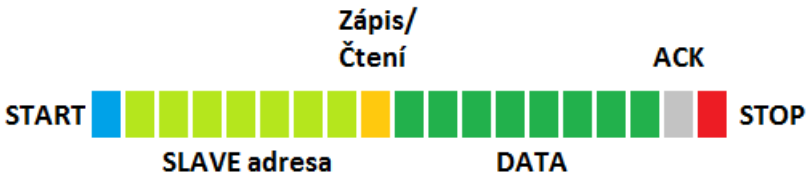
Programování mikrokontroléru ATmega2560

Pro naprogramování MCU byl použit USB kabel, který propojil USB konektor na AVR modulu a počítač. Po obdobných výpočtech propojek v internetovém kalkulátoru [21] (jako u MCU ATmega168 viz příloha B obrázek 4), s externím krystalem 16 Mhz, došlo k nastavení hodnot do softwaru AVRDUDE a výsledný kód byl připraven pro vložení do mikrokontroléru (viz příloha B obrázek 5).

Komunikace po sběrnici I²C

Tato sběrnice používá dva signálové vodiče. Prvním je *SDA*, který slouží pro obousměrný přenos dat. Druhý vodič *SCL* slouží k posílání hodinového signálu. Tyto dva vodiče musí být navíc připojeny přes zvedací rezistory k napájecímu napětí. V případě, že neprobíhá žádná komunikace, je na obou signálech díky těmto rezistorům logická "1"v pozitivní logice. Tento stav je normou stanovený klidový stav. Jak zařízení *Master*, tak zařízení *Slave* musí být připojeny ke stejné zemi (signál *GND*).

Sběrnice I²C komunikuje poloduplexně. To znamená, že v jeden okamžik vysílá pouze jedno zařízení a ostatní naslouchají. V této bakalářské práci byl použit jeden *Master* (ATmega2560) a jeden *Slave* (DS1338). Každý *Slave* má svoji danou adresu. I²C vlastní přesně stanovený komunikační protokol. Při zahájení komunikace nastaví *Master* na *SDA* logickou "0". *SCL* nechá ještě chvíli v logické "1"(záleží na nastavené přenosové rychlosti). Dále vybere jednu 7-bitovou adresu příslušného zařízení typu *Slave*(obr. 6.8). Následuje jeden bit, který určuje, zda-li chce *Master* data číst nebo zapisovat. Každý přenesený bajt je schválen potvrzovacím bitem ACK. Vše končí vzestupnou hranou *SDA* za předpokladu, že *SCL* již setrvává nějakou dobu v logické "1". V datasheetu firmy ATMEL nalezneme tuto komunikaci pod názvem TWI [5].



Obrázek 6.8: Přenos dat sběrnice I²C

Přerušení

Přerušení (anglicky interrupt) je schopnost procesoru okamžitě pozastavit dosavadně vykonávaný program za účelem vykonání speciální funkce. Zdroje přerušení se mohou dále dělit na vnitřní a vnější. Mezi vnitřní lze zařadit například přetečení časovače, příjem informace po určité sběrnici, dokončení převodu A/D převodníku. Za vnější zdroj přerušení se považuje změna stavu pinu procesoru.

U mikrokontroléru ATmega2560 bylo použito vnější přerušení na tlačítko pro návrat do menu, aby se nemuselo čekat na dokončení jednotlivých módů. Další použití se nalezlo i při přetečení několika časovačů např. pro piezosirénku, kde bylo vyžadováno přesné generování PWM signálu.

Čítače časovače

Časovač slouží k realizaci funkcí, u kterých je vyžadováno přesné generování signálu. Čítač bývá obvykle použit u opakujících se dějů, kde bývá zkoumána jejich periodicita. Obvykle bývají sjednoceny. V datasheetu mikročipu ATmega2560 lze například nalézt označení *Timer/Counter0* [5]. Až nastavením příslušného bitu v nastavovacím registru se tedy rozhodne, zda-li se bude jednat o čítač nebo časovač.

6.3 Druhá úloha

V některých výukových předmětech se vyskytovali studenti, kteří tuto lekci absolvovali jen okrajově a neměli moc zkušeností s programováním. Šlo zejména o studeny architektury. Právě pro ně byla navrhnuta tato část s použitím platformy Arduino.

6.3.1 Arudino specifikace

Arduino je open-source platforma, která používá mikrokontroléry z rodiny ATmega. Tento projekt vznikl v Itálii v roce 2005 ve městě Ivrea. Hlavní zakladatelé Massimo Banzi a David Cuartielles dali název tomuto projektu Arduino podle Arduina Ivrejského, který patří mezi významné historické postavy toho města. Výsledkem projektu měl být prototyp platformy pro studenty.

Programovací jazyk Arduina je založen na jazyku Wiring. Někdy se lze setkat s tvrzením, že je programováno právě v tomto jazyku, což není úplně pravda, protože malé odlišnosti se zde vykytují. Výhoda programovacího jazyku Arduina spočívá v široké distribuci kódů a knihoven na různých internetových stránkách, a proto je velice vhodný pro začínající programátory. Pro vytváření knihoven se používá programovací jazyk C++.

Arduino software

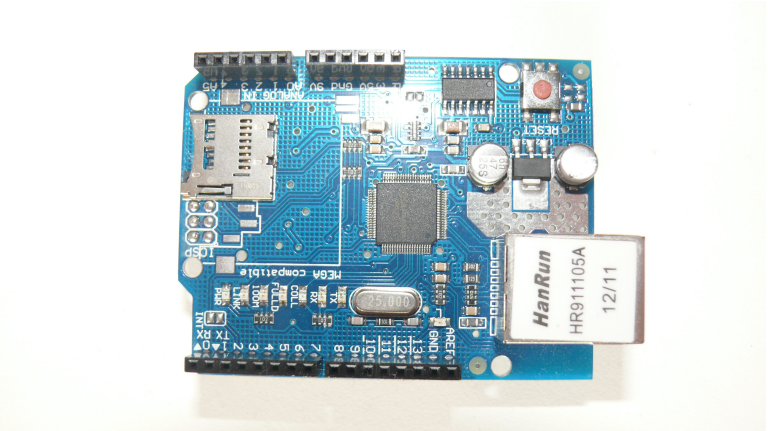
Open-source software byl stažen z domácí internetové stránky [13] pod názvem Arduino 1.0.5. Tento program je multiplatformní. Podporuje Windows, Linux i Mac OS X. Disponuje také integrovanými kompilátory: avr-gcc a avr-g++. Obsahuje základní knihovny pro jednotlivé Arduino Shieldy. Dále disponuje kompletní řadou bootloaderů pro hlavní Arduino desky (např. Arduino Uno, Arduino Mega 2560 atd.). Nahrát bootloader nebo samotný program umožňuje pětice programátorů, mezi kterými je na výběr i USBasp. Program také umožňuje velice jednoduché přidání a použití nových knihoven. Pro tento program je také dostupná čeština na internetových stránkách. V této bakalářské práci však nebyla použita. V neposlední řadě disponuje funkcí *Serial Monitor*. Při propojení USB portu mezi PC a libovolnou Arduino deskou dojde k vytvoření sériové komunikace. Použitím příslušné knihovny se pak jednotlivá hlášení v kódu mohou zobrazovat přímo v počítači a není potřeba žádného displeje.

Arduino Shield

Toto označení se používá pro jakýkoliv Arduino modul, který je možno zasunout do Ardunio Boardu (např. Arduino Uno, Arduino Mega 2560). Na internetu lze nalézt celou řadu Arduino Shieldů s nejrůznějšími periferiemi. Každý modul většinou disponuje svojí programovací knihovnou.

Při programování ukázkové úlohy byl použit Ethernet Arudino Shield [14]. Tento modul, jak se použito v názvu, umožňuje internetové připojení pomocí ko-

nektoru RJ45 (viz obr. 6.9). Vyveden je i slot pro microSD kartu. Jelikož modul obsahuje napájecí konektory, které jsou propojeny s Arduino deskou, odebírá napájecí napětí přímo z ní, tedy +5 V. Disponuje ethernetovým řadičem W5100 s interní vyrovnávací pamětí o hodnotě 16 kB. Rychlost připojení je 10/100 Mb/s. Komunikace s hlavní deskou probíhá po sběrnici SPI.



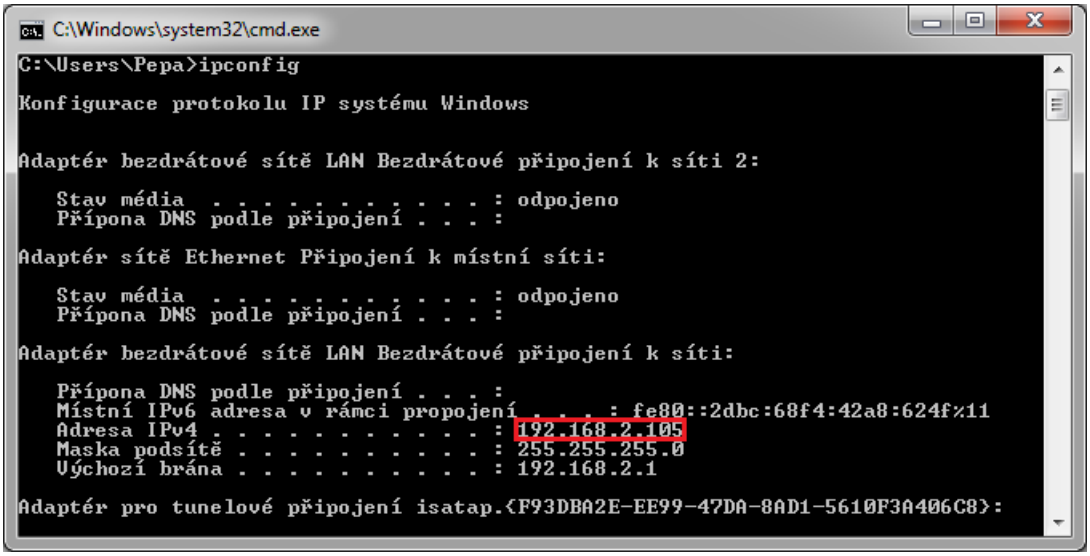
Obrázek 6.9: Ukázka Ethernet Arduino Shiledu

6.3.2 Funkce programu druhé úlohy

V této úloze došlo k vytvoření webového serveru, který ukazuje aktuální hodnotu potenciometru na přípravku v desetibitové hodnotě, tedy od 0 do 1023 (viz příloha B obrázek 6). Tato hodnota je také dále zobrazována na displeji výukového přípravku. Všechny hodnoty se zapisují na microSD kartu v půl vteřinovém intervalu (viz příloha B obrázek 7 a 11).

Při programování této úlohy byl využit příklad programu pro webový server. Následovalo vložení knihovny pro ethernet komunikaci a nastavení adresy MAC a IP. Arudino Ethernet Shiled nevlastní přidělenou MAC adresu. Byla tedy použita adresa ze starého modemu, který se již nepoužívá. IP adresa byla nastavena podle výpisu z příkazového řádku (viz obr. 6.10, str.42). Poslední tři čísla IP adresy však byla změněna na takové číslo, které žádná adresa v domácí síti neměla. Hodnota HTTP protokolu zůstala implicitně nastavena na hodnotu 80.

Dále bylo třeba vložit knihovnu pro displej a provést inicializaci. Pro správnou funkčnost se musely vložit piny, které odpovídali čtyřem datovým a třem řídícím vodičům. V inicializaci se pak nastavil počet znaků na jednom řádku displeje a celkový počet řádků (viz obr. 6.11, str. 42). Následovalo vložení poslední knihovny, která ovládala SD kartu a inicializace A/D převodníku pro pin potenciometru. Pro rozpoznání SD karty byl vložen kód, který při spuštění programu zjistí, o jaký typ karty přesně jde a jestli je karta přítomna v SD slotu. Jakmile není, program pokračuje bez možnosti zápisu hodnoty na kartu a na displeji se objeví hlášení *chyba zápisu*. Následuje načtení hodnoty pinu potenciometru. Dále se vytvoří server na



Obrázek 6.10: Zjištění IP adresy

```
LiquidCrystal lcd(37, 36, 35, 34, 33, 32, 31);
void init_lcd(void)
{
  // nastavení displeje, který obsahuje 16 znaků na dvou řádcích
  lcd.begin(16, 2);
}
```

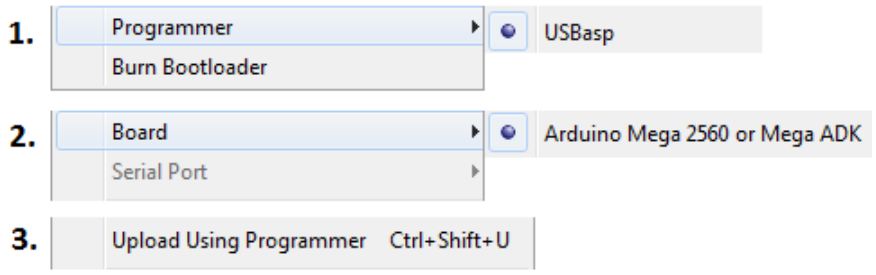
Obrázek 6.11: Nastavení displeje v jazyku Wiring

adrese *192.168.2.150*, který přijímá klienty a zobrazuje v HTTP hlavičce aktuální hodnotu potenciometru.

6.3.3 Programovací módy

Obecně jde jakákoliv Arduino Board programovat dvěma způsoby. První možností je programovat MCU pomocí programátoru (v tomto případě USBasp), který do něj nahraje mapu Arduino pinů pro příslušnou Arduino Board a vytvořený program. Návod (viz obr. 6.12, str. 43):

Druhý způsob využívá tzv. bootloaderu, což je v tomto případě aplikace, která umožňuje nahrát vytvořený program po sériové lince. Pro nahrání zavaděče je nastavení obdobné jako v prvním způsobu (obr. 6.12, str. 43), pouze dojde k vynechání třetího bodu. Místo toho se zvolí položka *Burn Bootloader* v sekci *Tools* a dojde k nahrávání pomocné aplikace do MCU. Tímto procesem přestává programování pomocí programátoru a nastává programování pomocí sériového portu(v tomto konkrétním případě propojení USB kabelu mezi počítačem a USB konektorem přípravku). Konečným krokem je nastavení správného portu v nástrojích *Tools*.



Obrázek 6.12: Nastavení přenosu programu do mikrokontroléru

Tímto krokem se otevře i funkce *Serial Monitor* (viz kapitola 6.3.1).

6.3.4 Nastavení pinů Arduino

Při programování mikrokontroléru ATmega2560 se společně s vytvořeným programem nahraje i rozložení Arduino pinů (viz příloha B obrázek 11). Každý pin má své číslo a je rozdělen na analogový nebo digitální. Při programování byla použita mapa pinů desky Arduino Mega 2560. Ta nepoužívá veškeré piny mikročipu. Modul AVR však ano, takže bylo nezbytné zasáhnout do adresáře *Arduino*, změnit knihovnu *pins_arduino.h* a důležité piny doplnit. Dodatečně vložené piny pak fungovaly jako klasické digitální piny.

7. Závěr

Řídícím mikrokontrolérem AVR modulu byl vybrán ATmega2560, který obsahoval vysoký počet programovatelných pinů, což bylo nezbytné pro ovládání potřebných periférií modulu i přípravku. Zvolené pouzdro TQFP100 vyhovuje ručnímu pájení, a přitom nezabírá velikou plochu DPS.

Pro návrh plošného spoje byl použit software KiCad, který by svojí vyspělostí mohl bez problému konkurovat mnoha podobným programům, za které je potřeba hradit licenční poplatky. Tento open-source software je určitě použitelný pro menší firmy nebo pro výuku na školách. Při návrhu DPS však došlo ke dvěma drobným chybám. První z nich bylo použití programovací propojky, která bohužel není v základní verzi schématu funkční. Při propojení konektoru JP2 totiž dochází ke sjednocení resetovacích pinů MCU obou mikrokontrolérů. Druhým nedostatkem je omezení programování Arduino kódu. Nefunguje druhý způsob nahrávání kódu pomocí sériové linky, kterou obstarává bootloader (viz kapitola 6.3.3). Oba tyto nedostatky byly napraveny vytvořením nového schématu (viz příloha C obrázek 4).

Před programováním jednotlivých úloh byl nejprve naprogramován mikrokontrolér ATmega168, který zastával funkci programátoru na AVR modulu. Díky této možnosti nebylo nutné trvale používat žádný externí programátor. Při programování tohoto modulu byl vytvořen program v jazyku C, který demonstruje skoro všechny periférie výukového přípravku. Naprogramovány byly také všechny komponenty AVR modulu. Dalším krokem bylo vyzkoušení Adruino platformy na tzv. Arduino Shield konektru, do kterého se zapojil speciální modul Ethernet Arduino Shied. Výsledkem byl funkční webový server, který zobrazoval aktuální desetibitovou hodnotu potenciometru na přípravku a následně ji ukládal do SD karty.

Výsledkem této práce je modul, který rozšířil použití výukového přípravku o další řadu mikrokontrolérů. Dalším přínosem tohoto modulu je možnost použití platformy Adruino, která svojí jednoduchostí umožní programování přípravku i začínajícím programátorům. Tento modul by šlo dále doplnit např. o sběrnici CAN a používat ho i v dalších předmětech, kde je tato sběrnice využívána.

Literatura

[1] Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, Morgan Skinner: C# 2008, Programujeme profesionálně, Computer Press, duben 2009, ISBN: 978-80-251-2401-7

[2] David Matoušek: Práce s mikrokontroléry Atmel AVR, BEN - technická literatura, červen 2006, ISBN: 80-7300-209-4

[3] FRÝZA, Tomáš. Mikroprocesorová technika a embedded systémy přednáška 1. *UREL* [online]. [cit. 12. května 2014]. Dostupné z: http://www.urel.feec.vutbr.cz/ fryza/downloads/mpt_pred_01.pdf

[4] PETERKA, Jiří. RISC vs. CISC. *eArchiv.cz* [online]. 2011 [cit. 12. května 2014]. Dostupné z: http://www.earchiv.cz/a92/a206c120.php3

[5] 8-bit Atmel Microcontroller with 64K/128K/256K Bytes In-System Programmable Flash. *Octopart* [online]. říjen 2012 [cit. 12. května 2014]. Dostupné z: http://datasheet.octopart.com/ATMEGA2560-16AU-Atmel-datasheet-14435218.pdf

[6] 8-bit Atmel Microcontroller with 4/8/16K Bytes In-System Programmable Flash. *Atmel* [online]. květen 2011 [cit. 12. května 2014]. Dostupné z: http://www.atmel.com/images/doc2545.pdf

[7] 8-bit Atmel Microcontroller with 16K Bytes In-System Programmable Flash. *Atmel* [online]. červenec 2010 [cit. 12. května 2014]. Dostupné z: http://www.atmel.com/Images/doc2466.pdf

[8] BALTIERI, Fabio. USB Key AVR Programmer. *fabiobaltieri* [online]. 2. září 2012 [cit. 12. května 2014]. Dostupný z: http://fabiobaltieri.com/2011/09/02/usb-key-avr-programmer/

[9] Serial programmer. *Projects in Embedded Systems* [online]. 30. května 2008 [cit. 12. května 2014]. Dostupný z: http://core.st/projects/Serial_programmer/

[10] 74HC595; 74HCT595 8-bit serial-in, serial or parallel out shift register with output latches; 3-state. *Embedded Linux Wiki* [online]. Nizozemsko, 25. května 2003 [cit. 12. května 2014]. Dostupné z: http://elinux.org/images/2/2d/74hc595.pdf

[11] DS1338 I2C RTC with 56-Byte NV RAM. *Maxim Integrated* [online]. Sunny Vale, CA, poslední aktualizace březen 2012 [cit. 12. května 2014]. Dostupné z: http://datasheets.maximintegrated.com/en/ds/DS1338-DS1338Z.pdf

[12] JAHELKA, Michal. Návrh elektroniky v programu KiCAD. *hw.cz* [online]. 13. května 2013 [cit. 12. května 2014]. Dostupné z: http://www.hw.cz/software/navrh-elektroniky-v-programu-kicad-8-dil.html

[13] Download the Arduino Software. *Arduino* [online]. 2014 [cit. 12. května 2014]. Dostupné z: http://arduino.cc/en/main/software

[14] Arduino Ethernet Shield. *Arduino* [online]. 2014 [cit. 12. května 2014]. Dostupné z: http://arduino.cc/en/Main/ArduinoEthernetShield

[15] SMT16030 DIGITAL TEMPERATURE SENSOR. *Smartec-sensors.com* [online]. Nizozemsko, červenec 2005 [cit. 12. května 2014]. Dostupné z:http://www.smartec.nl/pdf/DSSMT16030.PDF

[16] 5mm Red&Blue&Pure Green LED OSTA5131A-C. *Interelcom* [online]. [cit. 12. května 2014]. Dostupné z: http://www.interelcom.ru/files/doc/osta5131a-c.pdf

[17] PHOTOTRANZSISTOR KP-2012P3C. *tme.eu* [online]. 17. září 2001 [cit. 12. května 2014]. Dostupné z :http://www.tme.eu/cz/Document/ef1bf4df8516498bf28f69e2bac4ae07/kp-2012p3c.pdf

[18] Reflective Optical Sensor with Transistor Output. *Bwired.nl* [online]. Německo, 5. května 2000 [cit. 12. května 2014]. Dostupné z: http://www.bwired.nl/images/how/cny70.pdf

[19] Atmel Corporation © 2014. Atmel Studio. *Atmel* [online]. 2014 [cit. 12. května 2014]. Dostupné z: http://www.atmel.com/tools/atmelstudio.aspx

[20] AVRDUDE. *Savannah* [online]. 8. ledna 2010 [cit. 12. května 2014]. Dostupné z: http://www.nongnu.org/avrdude/

[21] Engbedded Atmel AVR® Fuse Calculator. *Engbedded ensouling circuits* [online]. [cit. 12. května 2014]. Dostupné z: http://www.engbedded.com/fusecalc

[22] ZHAO, Frank. AVR Timer Calculator. *ELECCELERATOR* [online]. 27. září 2013 [cit. 13. května 2014]. Dostupné z: http://eleccelerator.com/avr-timer-calculator/

[23] Programming Device Manual BookletAVR Prog USB v2. *and-tech.pl* [online]. Polsko, [cit. 12. května 2014]. Dostupné z: http://and-tech.pl/wp-content/download/avrprog

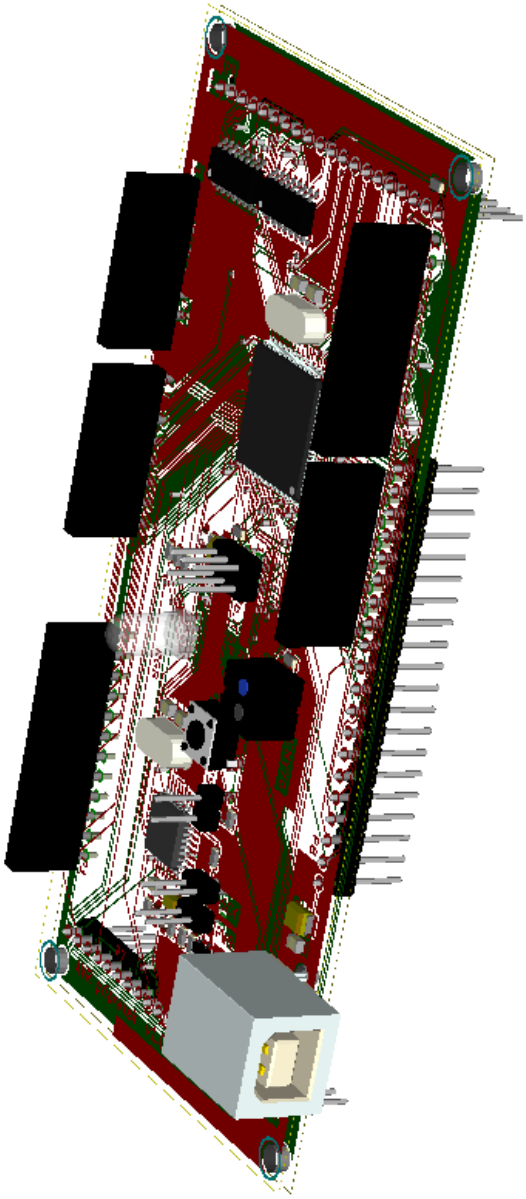
[24] ATmega2560-Arduino Pin Mapping *Arduino* [online]. 2014 [cit. 13. května 2014]. Dostupné z: http://arduino.cc/en/Hacking/PinMapping2560

A. Obsah přiloženého CD

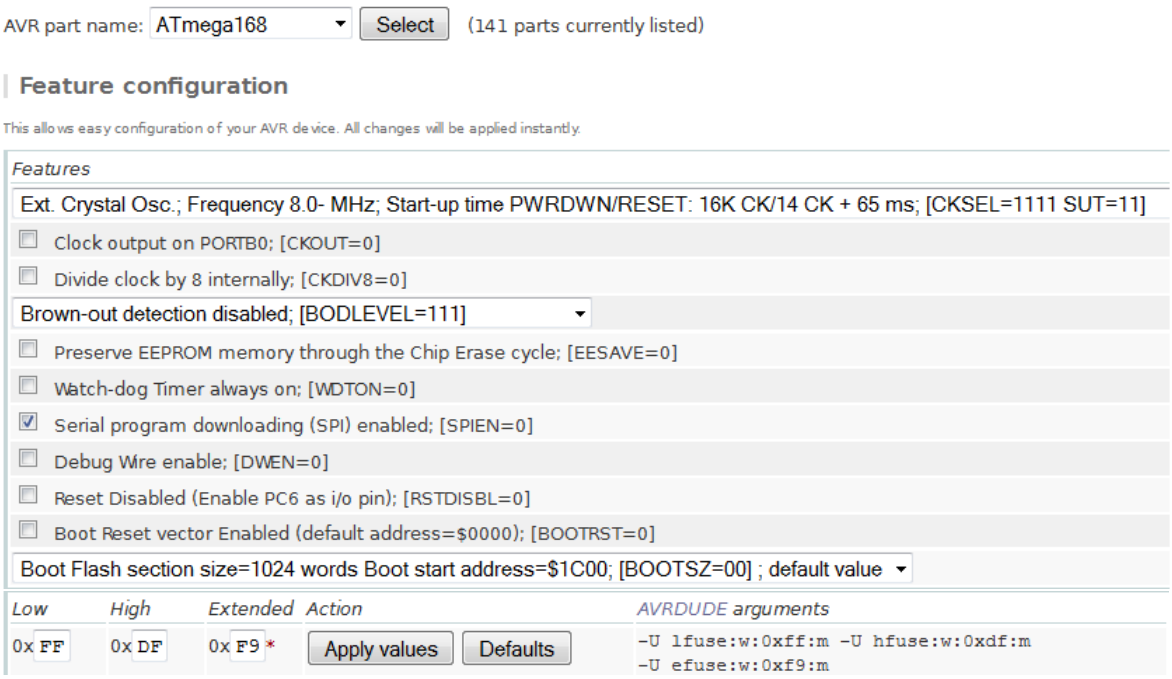
Obsah bakalářské práce byl rozdělen do těchto složek:

- Text_BP - text bakalářské práce ve formátu PDF
- Obrázky - obrázky použité v bakalářské práci
- Arduino - kód druhé úlohy, instalace vývojového prostředí a mapa pinů Arduino Mega 2560
- Návrh_DPS - schéma a layout AVR modulu, seznam součástek, instalace programu Kicad, použité knihovny, výrobní soubory (formát Gerber a Excellon)
- Avrdude - instalace programu a grafické rozhraní
- Kód první úlohy - projekt první úlohy v Atmel Studiu a použité knihovny a podprogramy
- USBasp - kód pro ATmega168 nebo ATmega8 a ovladače

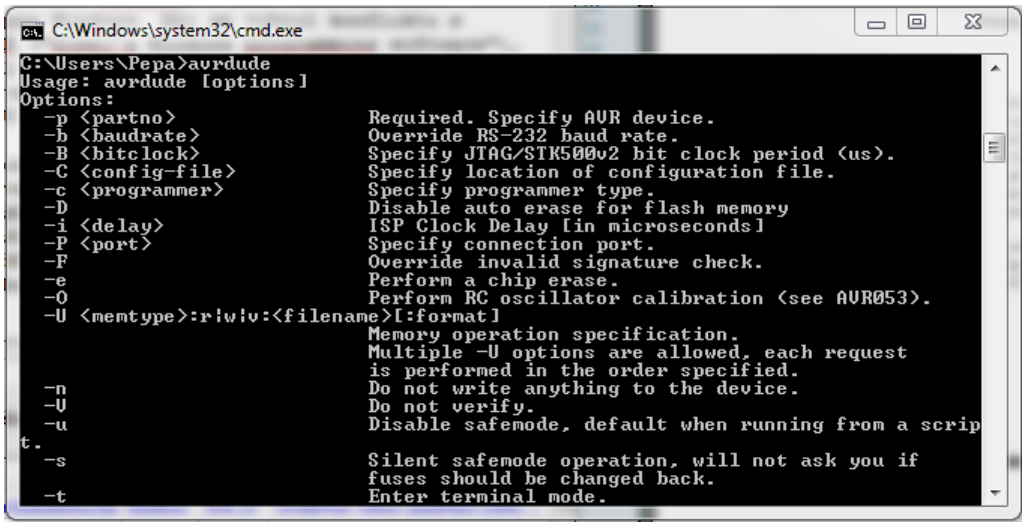
B. Ostatní obrázky



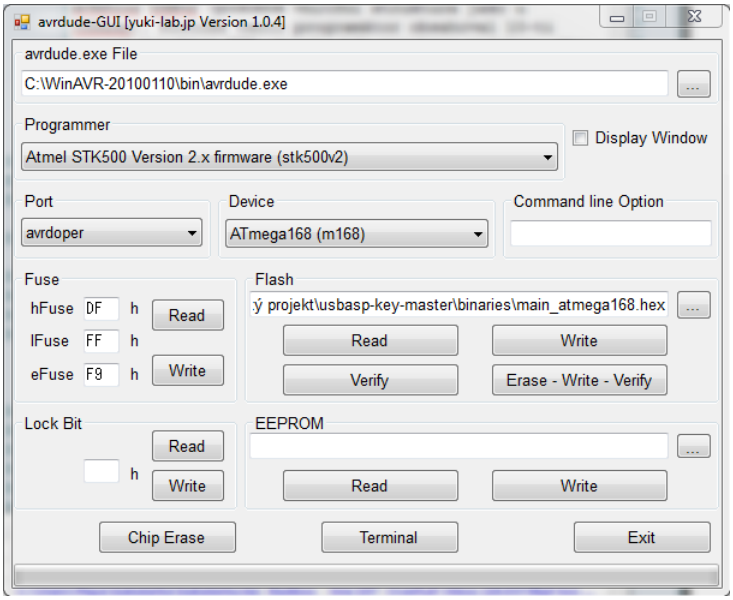
Obrázek B.1: 3D vizualizace AVR modulu



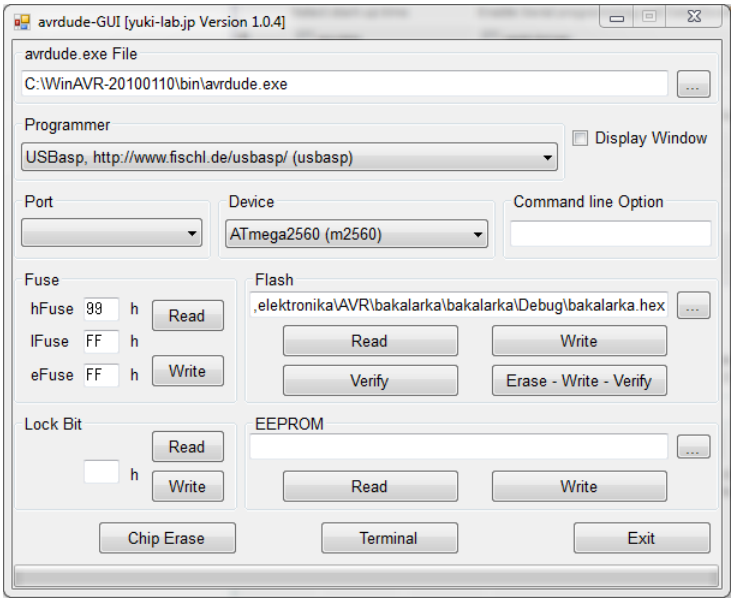
Obrázek B.2: Nastavení pojistek pro USBasp programátor



Obrázek B.3: Programování MCU v avdude v příkazovém řádku

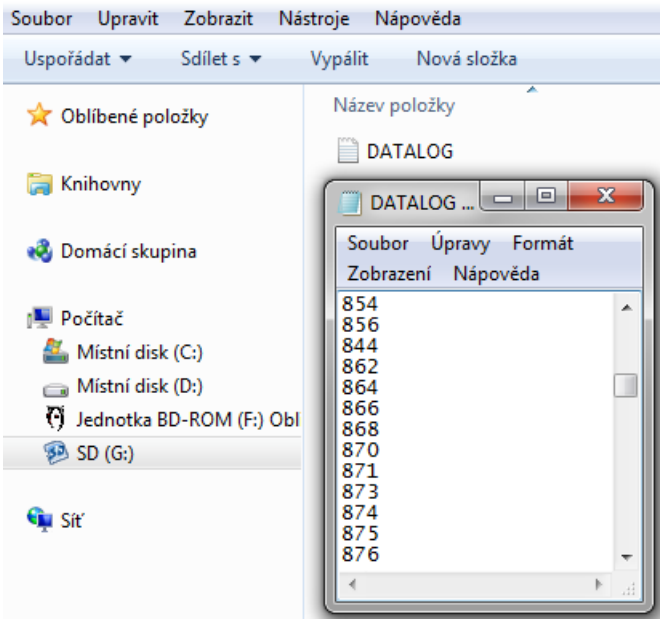


Obrázek B.4: Nastavení mikrokontroléru ATmega168 v avrdude-GUI

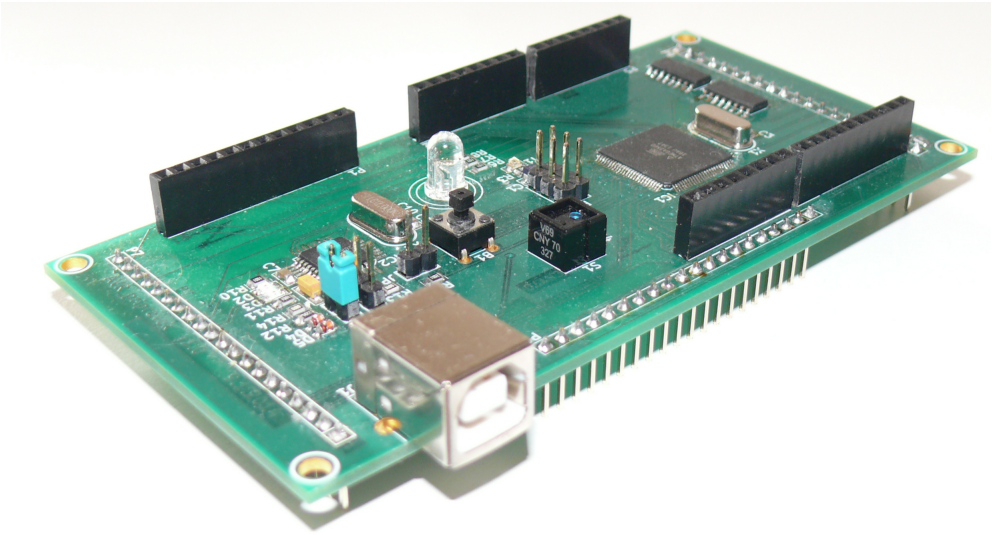


Obrázek B.5: Nastavení mikrokontroléru ATmega2560 v avrdude-GUI

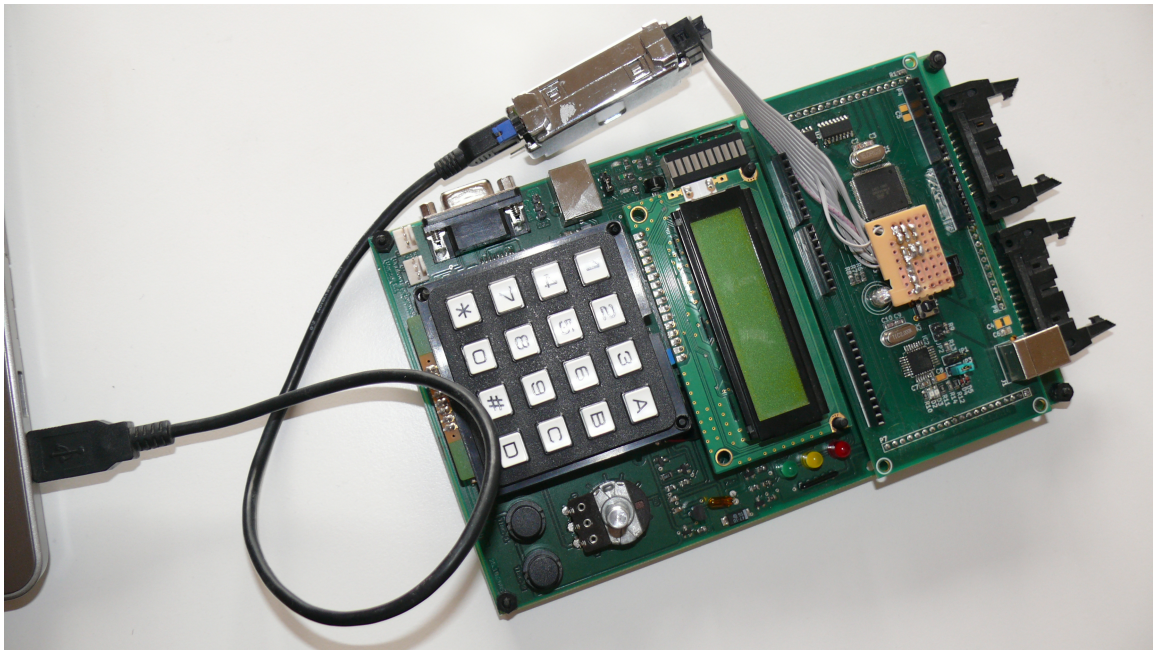
Obrázek B.6: Ukázka funkce webového serveru



Obrázek B.7: Zobrazení dat na SD kartě

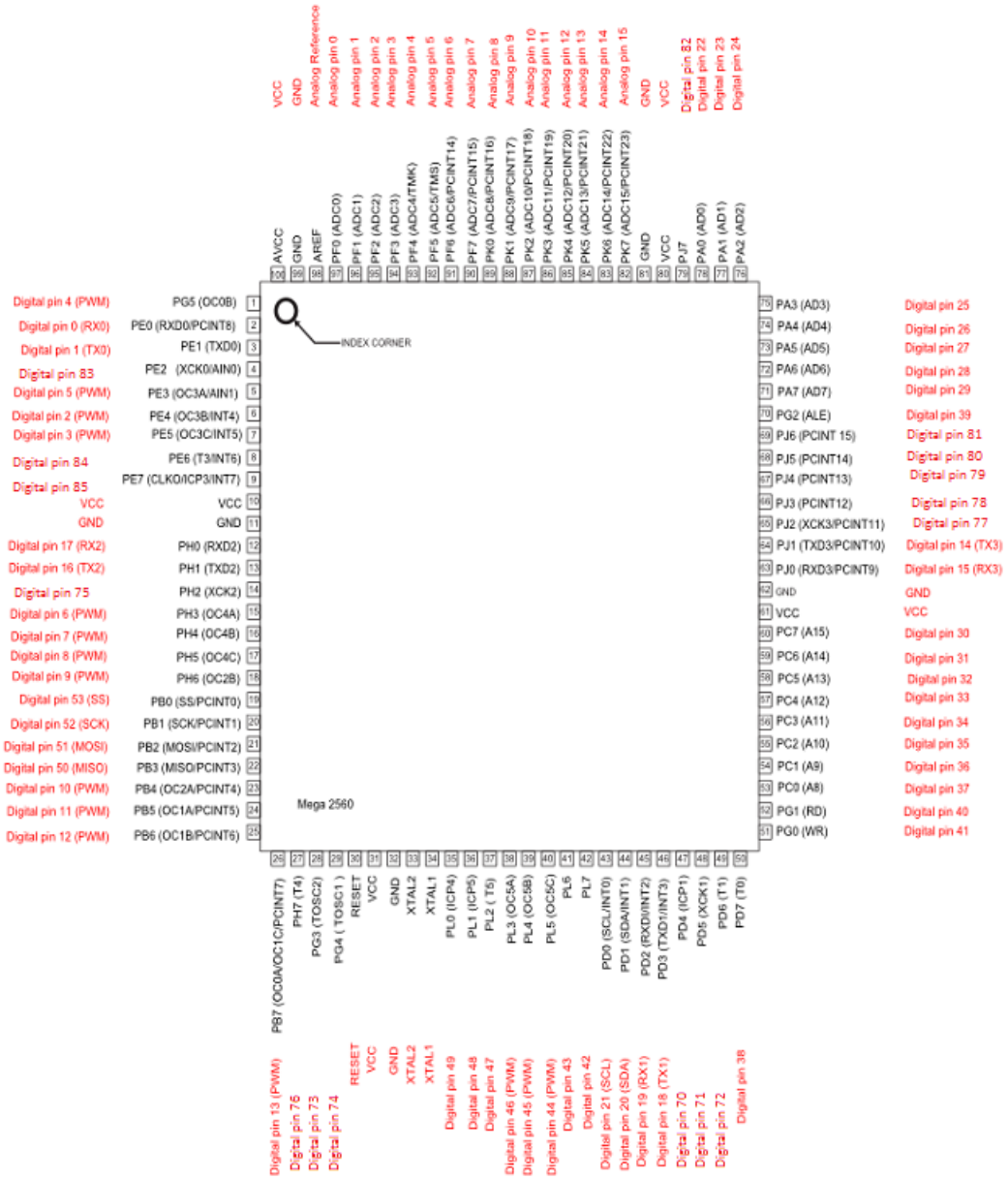


Obrázek B.8: Hotový AVR modul



Obrázek B.9: Programování ATmega168 pomocí AVRProg programátoru

Obrázek B.10: Připojený Arduino Shield k routeru



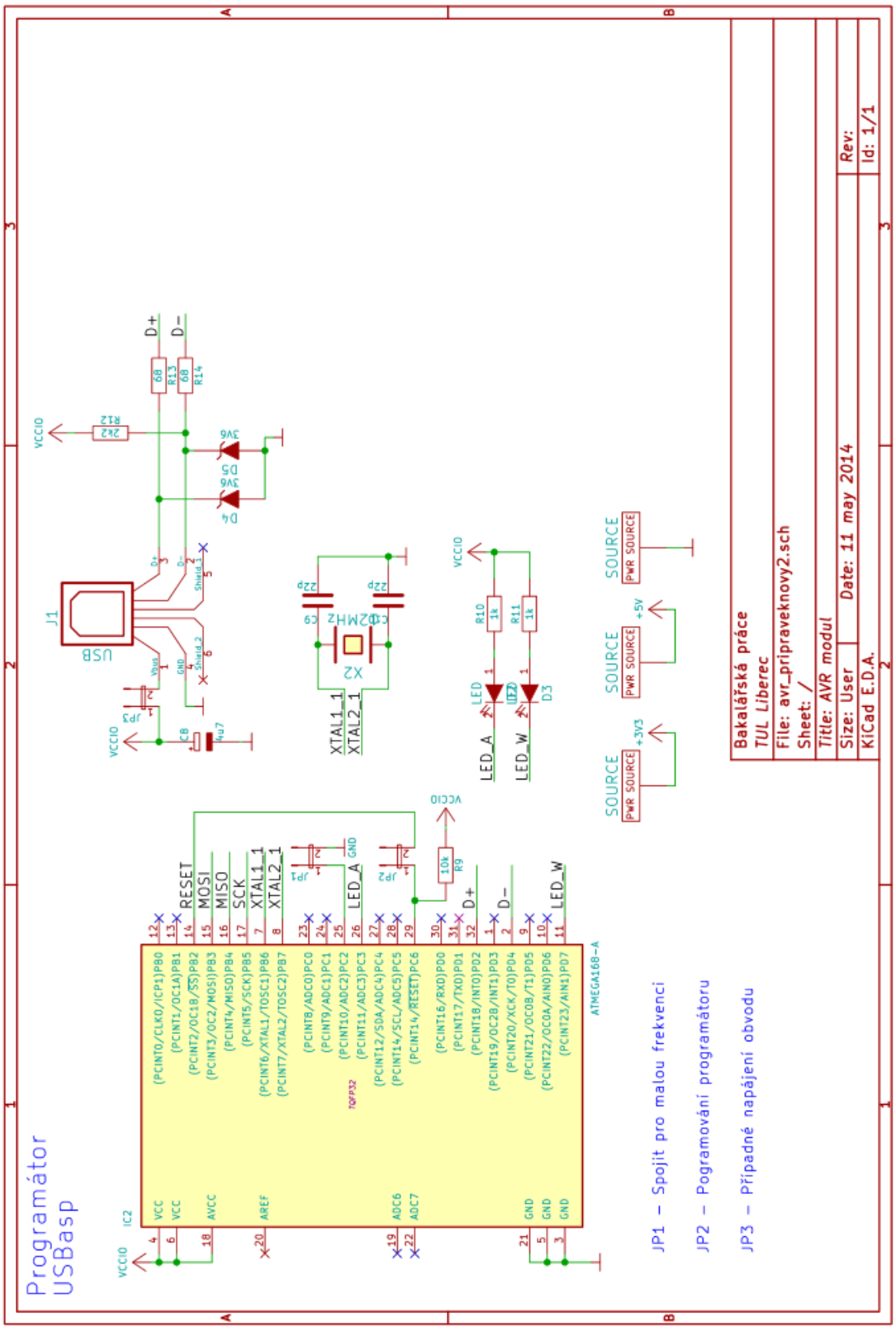
Obrázek B.11: Mapa pinů pro Arduino Mega 2560 pinout

Obrázek byl převzat z [24] a následně upraven.

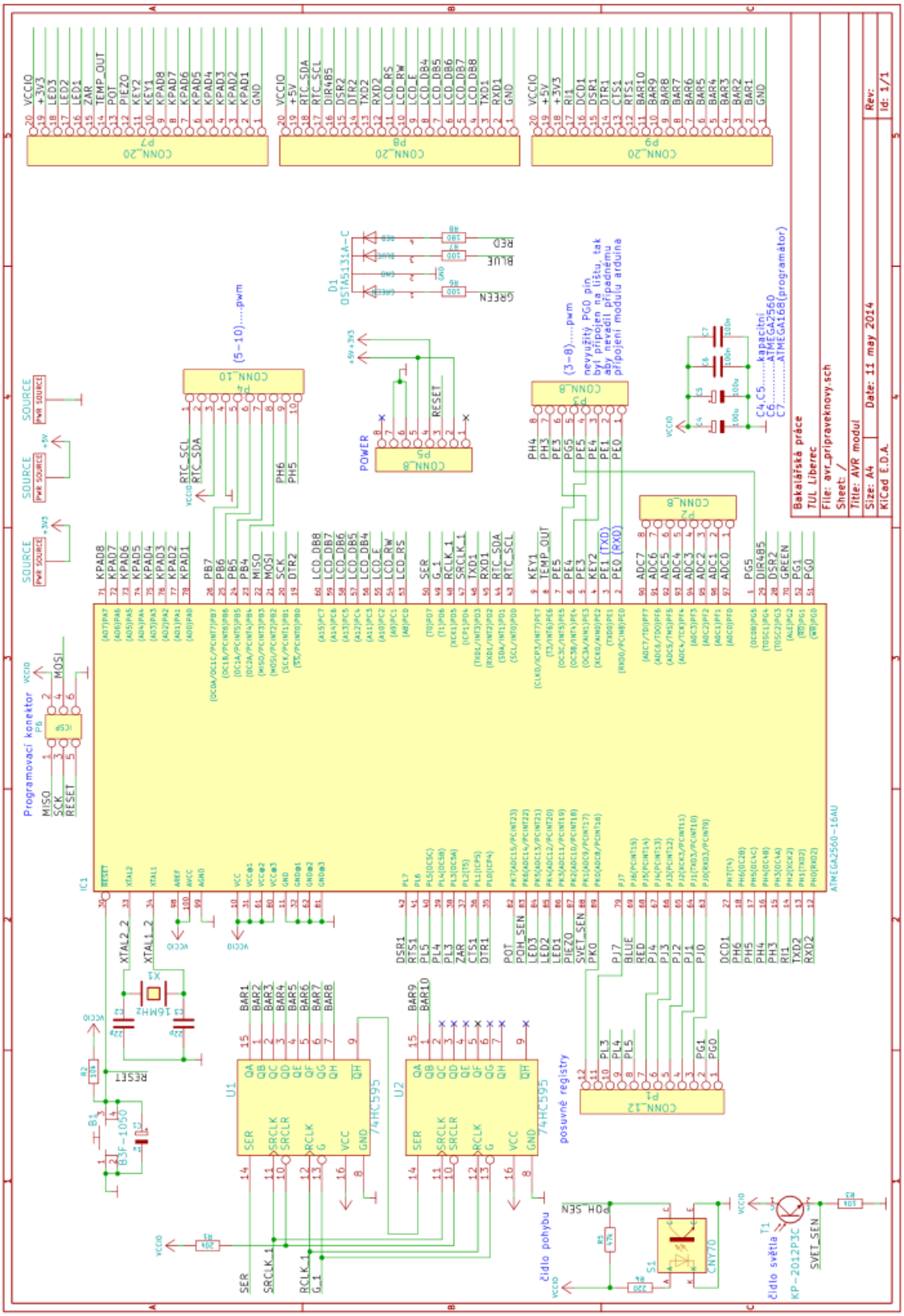
C. Schéma, layout a součástky DPS

ID	Určovatel	Pouzdro	Množství	Hodnota
1	P1	SIL-12	1	CONN_12
2	D1	RGB	1	OSTA5131A-C
3	P5,P3,P2	SIL-8	3	CONN_8
4	P6	SIL-3x2	1	ICSP
5	IC1	TQFP_100	1	ATMEGA2560-16AU
6	J1	USB_B	1	USB B
7	S1	CNY70	1	CNY70
8	X2	HC-49V	1	12MHz
9	X1	HC-49V	1	16MHz
10	P4	SIL-10	1	CONN_10
11	JP2,JP3,JP1	SIL-2	3	JP2
12	P7,P8,P9	SIL-20	3	CONN_20
13	B1	BTN4.5x6.5	1	B3F-1050
14	IC2	TQFP32	1	ATMEGA168-A
15	R6,R7	SM0805	2	100
16	R4	SM0805	1	220
17	R3,R9,R2	SM0805	3	10k
18	R5	SM0805	1	47k
19	R10,R11	SM0805	2	1k
20	R1	SM0805	1	20k
21	R14,R13	SM0805	2	68
22	R12	SM0805	1	2k2
23	R8	SM0805	1	180
24	D5,D4	SOD323	2	3V6
25	U2,U1	SO16	2	74HC595
26	C9,C3,C2,C10	SM0805	4	22p
27	C7,C6	SM0805	2	100n
28	C5,C4	c_tant_B	2	100u
29	C1	c_tant_A	1	1u
30	C8	c_tant_A	1	4u7
31	D2,D3	SM0805	2	LED
32	T1	SM0805	1	KP-2012P3C

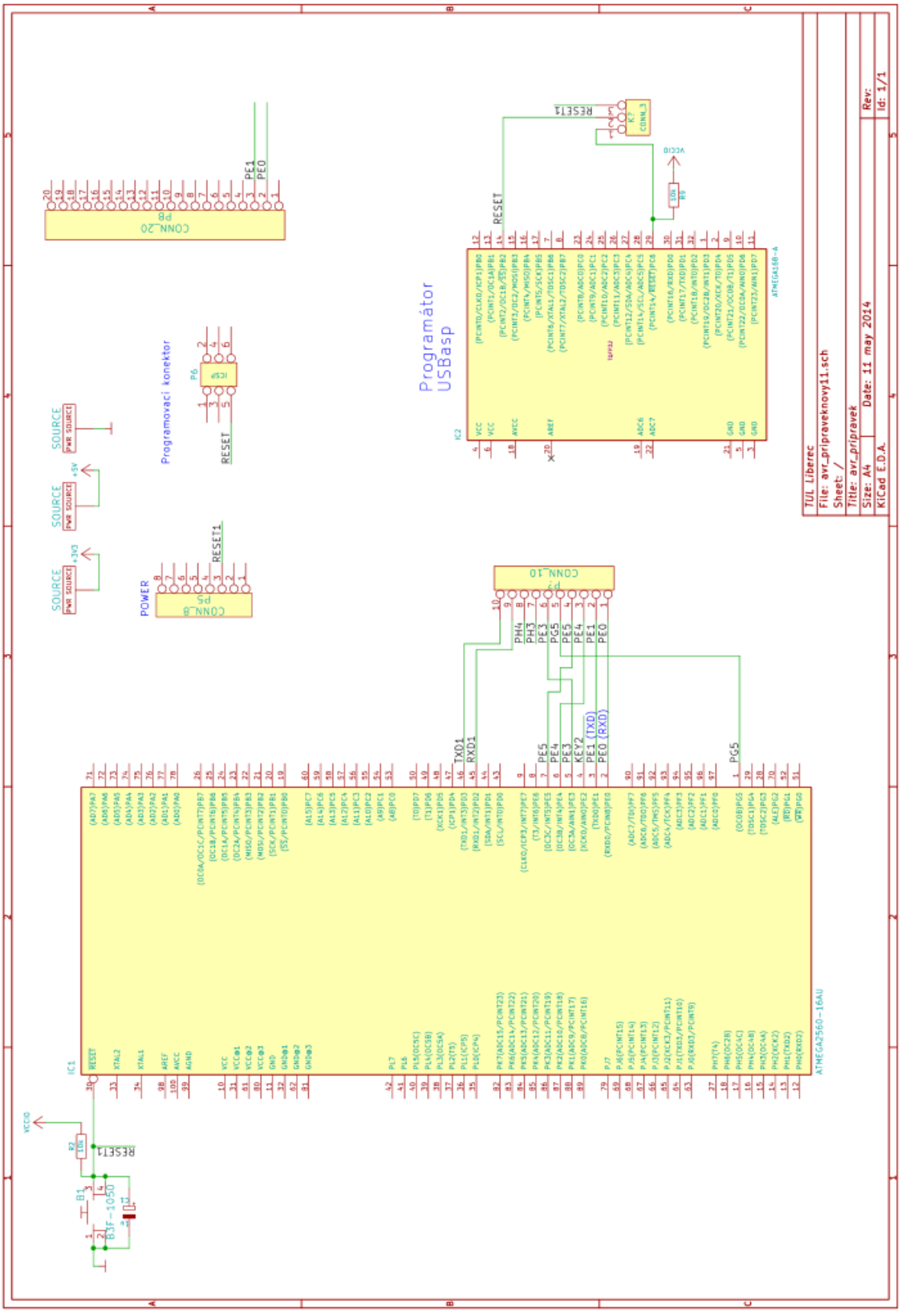
Obrázek C.1: Seznam součástek AVR modulu



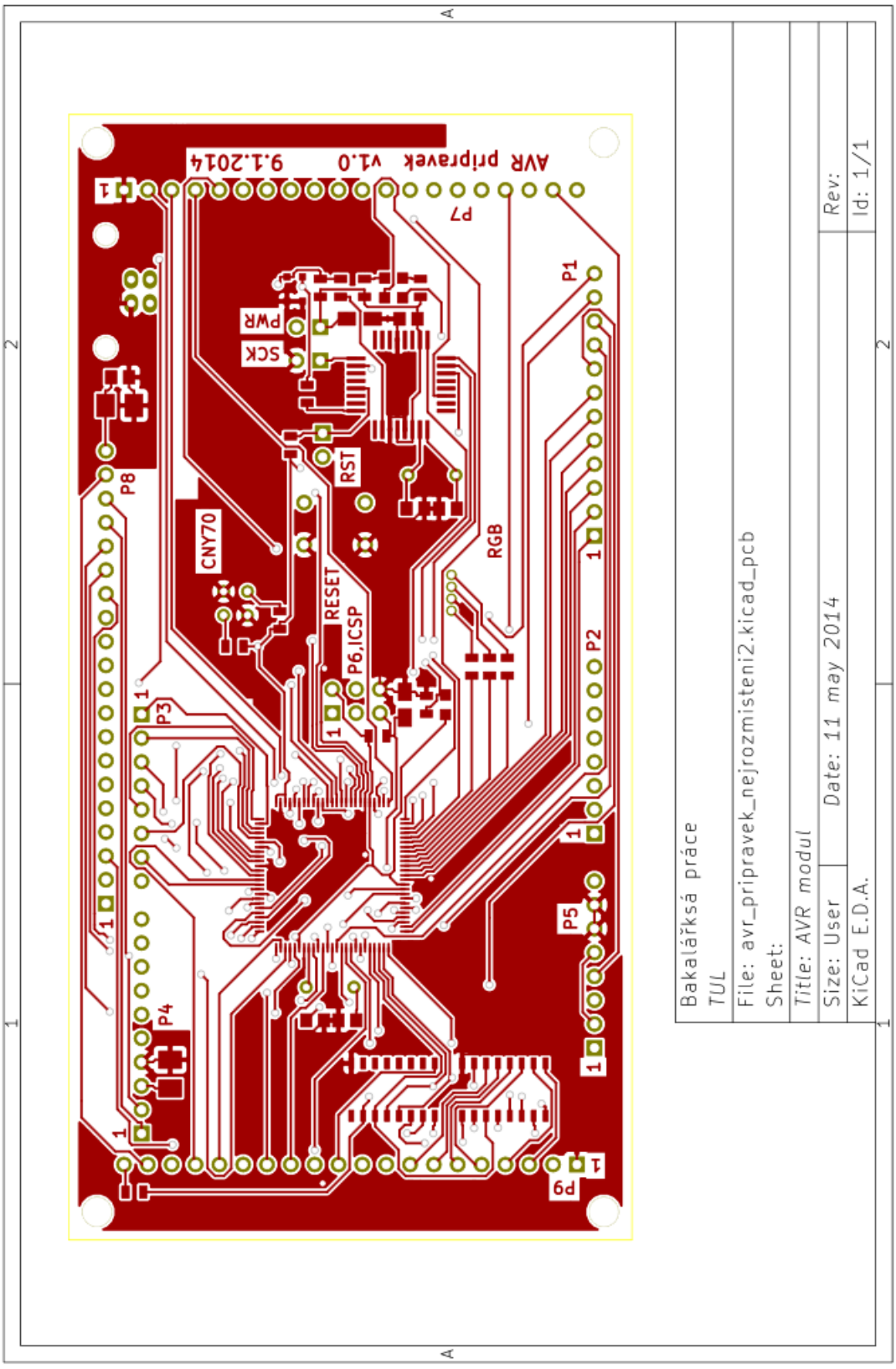
Obrázek C.2: Schéma programátoru USBasp na AVR modulu



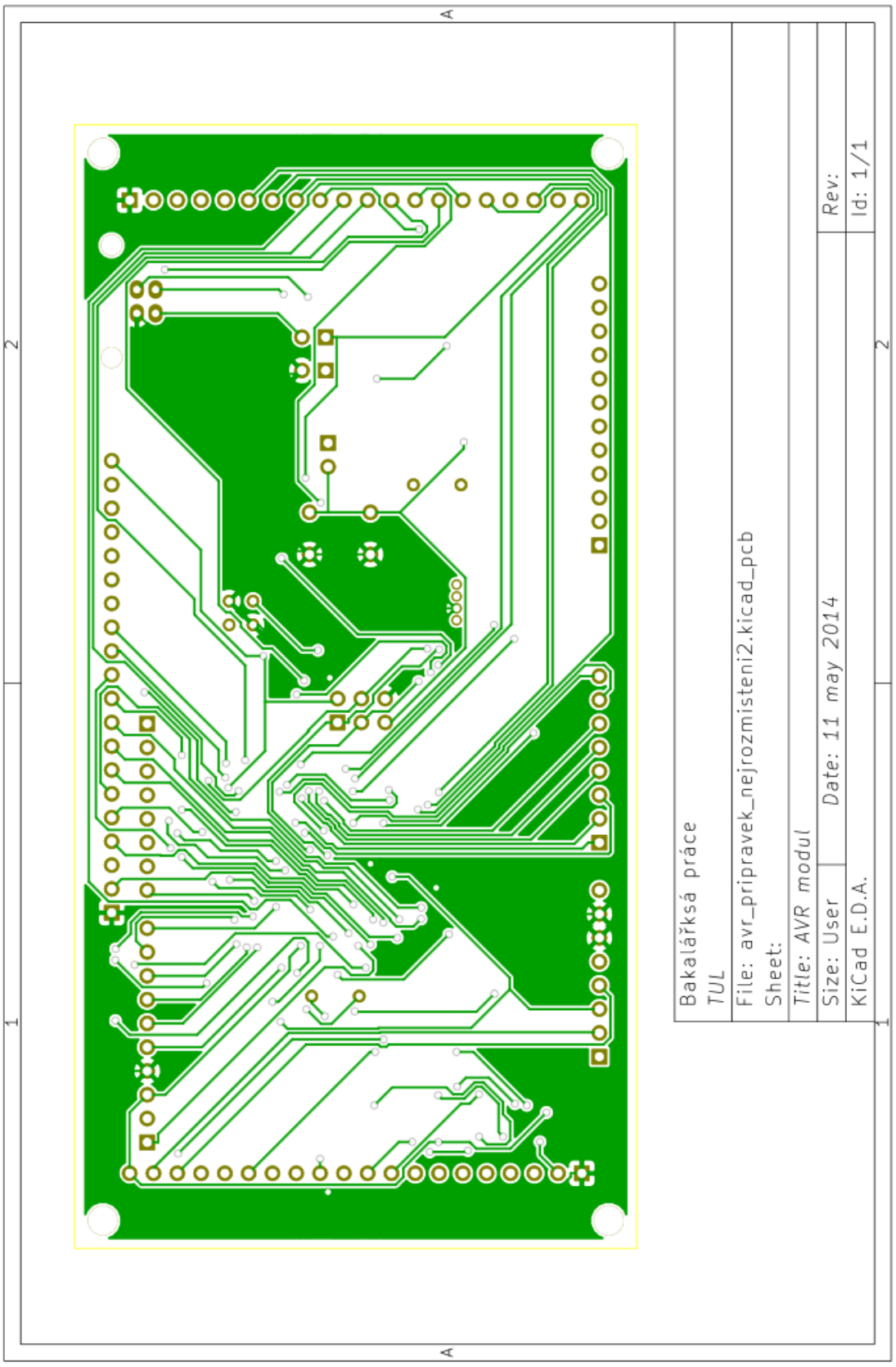
Obrázek C.3: Schéma řídícího obvodu AVR modulu



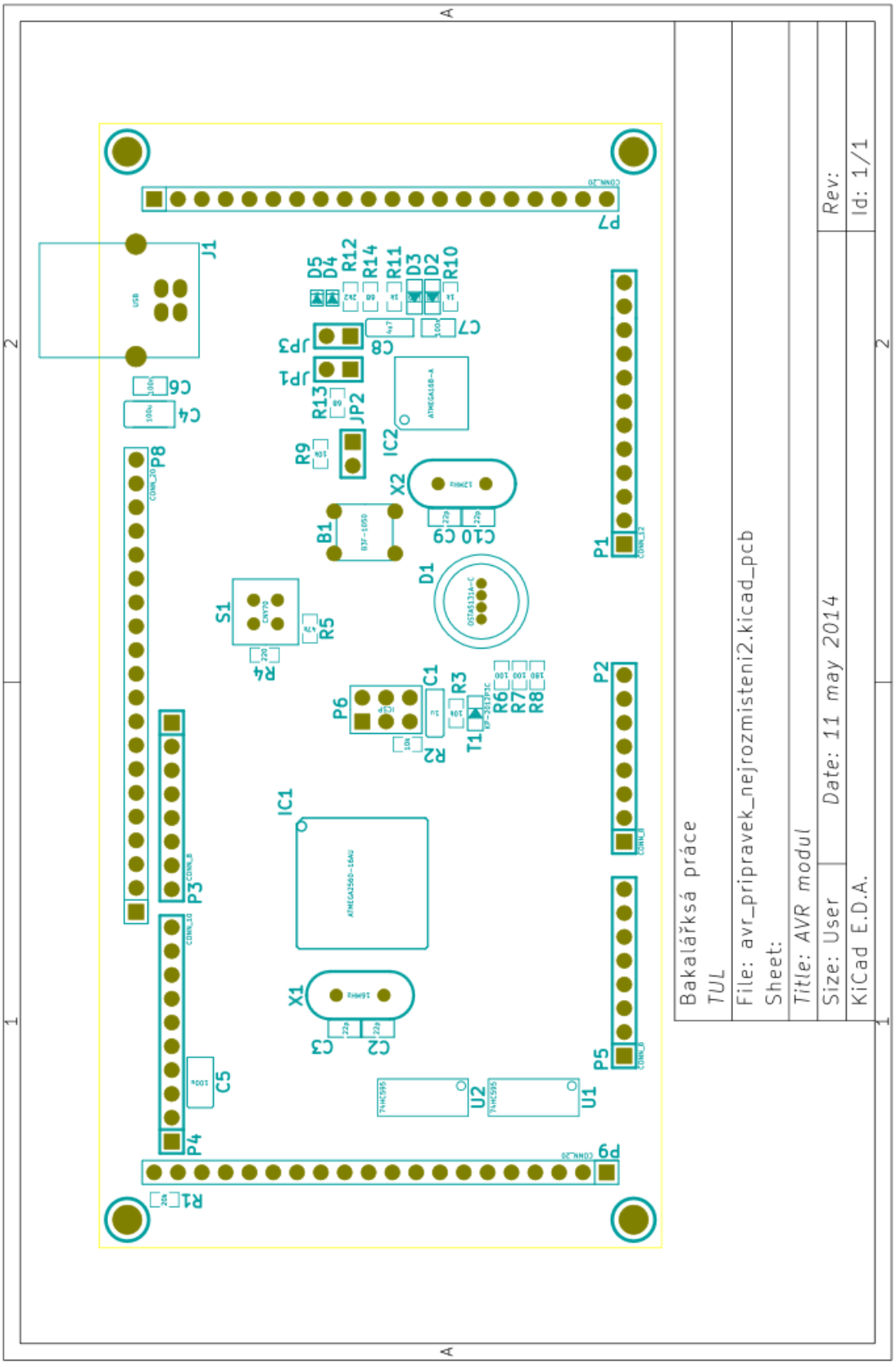
Obrázek C.4: Schéma úprav AVR modulu pro další verzi



Obrázek C.5: Layout svrchní vrstvy mědi AVR modulu



Obrázek C.6: Layout spodní vrstvy mědi AVR modulu



Obrázek C.7: Umístění součástek na svrchní vrstvě AVR modulu